

PCT
526 Rec'd PCT/PTC 17 AUG 2000

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#4

Applicant(s): Jae-Yoel KIM et al.

Docket: 678-511

Serial No.: 09/611,069

Dated: August 15, 2000

Filed: July 6, 2000

For: APPARATUS AND METHOD FOR ENCODING/
DECODING TRANSPORT FORMAT
COMBINATION INDICATOR IN CDMA
MOBILE COMMUNICATION SYSTEM

RECEIVED
FEB 05 2001
Technology Center 2600

Assistant Commissioner for Patents
Washington, D.C. 20231

TRANSMITTAL OF PRIORITY DOCUMENT

Sir:

Enclosed is a certified copy of Korean Appln. No. 27932/1999 filed on
July 6, 1999 and from which priority is claimed under 35 U.S.C. §119.

Respectfully submitted,

Paul J. Farrell
Registration No. 33,494
Attorney for Applicants

DILWORTH & BARRESE
333 Earle Ovington Boulevard
Uniondale, New York 11553
(516) 228-8484

PJF/MM:mg

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8(a)

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postpaid in an envelope, addressed to the: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on August 15, 2000.

Dated: August 15, 2000

Michael J. Musella

P427-43



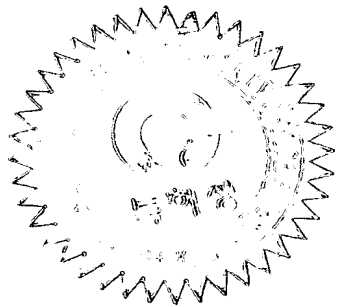
별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Industrial Property Office.

출원 번호 : 특허출원 1999년 제 27932 호
Application Number

출원 년 월 일 : 1999년 07월 06일
Date of Application

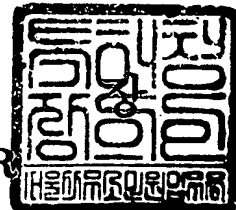
출원인 : 삼성전자 주식회사
Applicant(s)



2000 년 07 월 26 일

특 허 청

COMMISSIONER



【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	1999.07.06
【국제특허분류】	H03M
【발명의 명칭】	부호분할다중접속시스템의 부호화/복호화장치 및 방법
【발명의 영문명칭】	APPARATUS AND METHOD FOR CODING AND DECODING IN CDMA COMMUNICATION SYSTEM
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	1999-006038-0
【발명자】	
【성명의 국문표기】	김재열
【성명의 영문표기】	KIM, Jae Yoel
【주민등록번호】	700219-1047637
【우편번호】	435-042
【주소】	경기도 군포시 산본2동 산본9단지 백두아파트 960동 1401호
【국적】	KR
【발명자】	
【성명의 국문표기】	강희원
【성명의 영문표기】	KANG, Hee Won
【주민등록번호】	680119-1051636
【우편번호】	131-207
【주소】	서울특별시 중랑구 면목7동 1499번지 용마 동아아파트 10동 902호
【국적】	KR
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 주 (인)

【수수료】

【기본출원료】 20 면 29,000 원

【가산출원료】 66 면 66,000 원

【우선권주장료】 0 건 0 원

【심사청구료】 0 항 0 원

【합계】 95,000 원

【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 부호분할다중접속시스템의 QoF를 이용한 TFCI부호화 및 복호화 장치 및 방법에 관한 것으로, m 시퀀스를 열치환 함수에 의해 열치환하여 기저 마스크함수를 생성하고, 상기 생성된 마스크함수와 월시부호를 이용하여 TFCI 정보비트를 마스크함수로 부호화를 행하고, 복호화시 수신신호를 기저 마스크함수로 승산한 후 월시부호와의 상관치에 의해 복호화를 수행하는 장치 및 방법을 구현하였다.

【대표도】

도 8

【색인어】

QoF, TFCI, 부호화, 복호화

【명세서】**【발명의 명칭】**

부호분할다중접속시스템의 부호화/복호화장치 및 방법{APPARATUS AND METHOD FOR CODING AND DECODING IN CDMA COMMUNICATION SYSTEM}

【도면의 간단한 설명】

도 1은 부호분할다중접속시스템에서 TFCI의 사용 예를 보이기 위한 프레임 구조를 도시한 도면.

도 2는 프레임을 전송하기 위한 통상적인 기지국 송신기의 구조를 개시한 도면.

도 3은 프레임을 전송하기 위한 통상적인 이동국 송신기의 구조를 개시한 도면.

도 4는 종래 기본적인 TFCI의 오류정정부호화 방식에 따른 구성을 도시한 도면.

도 5는 종래 확장된 TFCI의 오류정정부호화 방식에 따른 구성을 도시한 도면.

도 6은 본 발명의 일 실시 예에 따른 기본적인 TFCI와 확장된 TFCI 구조의 오류정정부호화 방식에 따른 송신기의 개념적인 구성을 도시한 도면.

도 7은 본 발명의 일 실시 예에 따른 마스크함수를 생성하기 위한 알고리즘을 개시하고 있는 도면.

도 8은 본 발명의 일 실시 예에 따른 송신기를 구성하는 TFCI부호화기의 구조를 도시한 도면.

도 9는 본 발명의 일 실시 예에 따른 수신기를 구성하는 TFCI복호화기의 구조를 도시한 도면.

도 10은 도 9에서 개시하고 있는 상관도 비교기에서 수행하는 제어 흐름을 도시한 도면.

도 11은 본 발명의 일 실시 예에 따른 부호화기에서 수행하는 제어 흐름을 도시한 도면.

도 12는 본 발명의 다른 실시 예에 따른 부호화기에서 수행하는 제어 흐름을 도시한 도면.

도 13은 본 발명의 실시 예에 따른 TFCI 비트에 대응하는 사용되는 월시부호 및 마스크함수를 도시하고 있는 도면.

도 14는 본 발명의 다른 실시 예에 따른 송신기를 구성하는 TFCI부호화기의 구조를 도시한 도면.

도 15는 본 발명의 다른 실시 예에 따른 수신기를 구성하는 TFCI복호화기의 구조를 도시한 도면.

도 16은 도 14에서 개시하고 있는 부호화기에서 수행하는 제어 흐름을 도시한 도면

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<17> 본 발명은 부호분할다중접속시스템의 전송을 정보(Transport Format Combination Indicator, 이하 TFCI라 칭한다)부호발생기에 관한 장치 및 방법에 관한 것으로, 특히

QoF를 이용한 TFCI부호화 및 복호화 장치 및 방법에 관한 것이다.

<18> 일반적으로 부호분할을 수행하는 이동통신시스템(이하 IMT2000 시스템이라 칭한다)에서는 하나의 물리적 채널 내에 음성서비스, 화상서비스, 데이터서비스와 같은 여러 가지 서비스의 프레임들을 같이 전송한다. 상기 서비스들의 프레임은 고정된 데이터 전송율로 전송되거나 가변적인 전송율로 전송된다. 고정된 전송율로 전송되는 서로 다른 서비스들은 각각의 서비스의 프레임의 확산레이트(spreading rate)를 수신측에 일일이 알려줄 필요가 없다. 하지만, 가변적인 전송율로 전송되는 서비스들은 데이터 전송율이 서비스 중간에 달라질 수 있으므로 각 서비스 프레임의 전송율에 의해 결정되는 확산레이트를 수신측에 알려줘야 한다. 이때, 통상적인 IMT2000 시스템에서는 데이터의 전송속도와 데이터의 확산레이트가 반비례한다. 상기 각각의 서비스가 사용하는 프레임의 전송속도가 다른 경우에 현재 전송되고 있는 서비스의 조합을 알려주는 역할 하는 것이 TFCI이다. 상기 TFCI는 각각의 전송율에 따른 서비스가 올바르게 이루어질 수 있도록 한다. 상기 TFCI의 사용 예는 도 1과 같다. 상기 도 1에서도 개시하고 있는 바와 같이 상기 TFCI의 값은 길이 32인 부호로 부호화되어 각 T slot($0.625\mu s$)마다 2비트씩 삽입되며, 10ms 무선 프레임(Radio Frame)에는 총 32비트로 부호화된 TFCI의 값이 입력되어 전송된다.

<19> 도 2는 프레임을 전송하는 통상적인 기지국 송신기의 구조를 도시한다.

<20> 상기 도2를 참조하면, 곱셈기 211, 221, 231, 232는 멀티플렉서 201, 202, 203, 204에 의해 각각 멀티플렉싱된 신호를 받아 이득계수 G_1 , G_2 , G_3 , G_4 를 곱하기 위한 장치이다. 상기 이득계수 G_1 , G_2 , G_3 , G_4 는 서비스 종류나 핸드오버 상황 등과 같이 상황에 따라 각기 다른 값을 가질 수도 있다. 한편, 상기 멀티플렉서 201, 202, 203, 204는 각

각의 채널에 채널부호화 및 인터리빙을 수행한 DPCCH 및 DPDCH 신호나 파일럿 신호, 전력제어 신호등으로 이루어지는 기타신호들이 입력되면 상기 입력되는 기타신호들과 각 TFCI 부호기로부터 제공되는 32비트의 부호화 심볼들을 멀티플렉싱하여 상기 곱셈기 211, 221, 231, 232로 각각 출력한다. 상기 32비트의 부호화 심볼들은 각 채널의 데이터 신호들에 따른 전송속도 정보 등을 정의하는 TFCI비트가 TFCI 부호기에 입력되어 얻어지는 심볼들이다. 멀티플렉서 212는 DPCCH와 DPDCH를 시간적으로 멀티플렉싱하여 상기 도 1과 같은 슬롯구조를 가지도록 한다. 제1 S/P 213은 직병렬변환기로써 상기 멀티플렉서 212의 출력을 I채널과 Q채널로 분배하기 위한 장치이다. 제2 S/P 233과 제3 S/P 234는 DPDCH₂ 및 DPDCH₃을 직-병렬 변환하여 I채널과 Q채널로 분배하기 위한 장치이다. 상기 직-병렬 변환된 출력들은 곱셈기 214, 222, 235, 236, 237, 238로 각각 공급되어 확산 및 채널구분을 위해 채널구분코드(Channelisation code) C_{ch1} , C_{ch2} , C_{ch3} 와 곱해진다. 상기 채널구분코드는 직교부호가 사용된다. 상기 곱셈기 214, 235, 237에서 채널구분코드와 곱해진 출력은 제1합산기 215에서 합산되어 I채널 신호로 생성되며, 상기 곱셈기 222, 236, 238에서 채널구분코드와 곱해진 출력은 제2합산기 223에서 합산되어 Q채널 신호로 생성된다. 상기 제2합산기 223에 의해 합산된 Q채널 신호는 위상천이기 224에서 90도 위상이 바뀐다. 합산기 216은 제1 합산기 215의 출력과 위상천이기 224의 출력을 합산하여 복소신호 $I+jQ$ 신호를 생성한다. 곱셈기 217은 상기 복소신호를 각 기저국별로 할당된 PN시퀀스(C_{scramb})에 의하여 스크램블링하며, 제4 S/P 218에서는 상기 스크램블링된 신호를 직-병렬 변환하여 I채널 및 Q채널로 분배한다. 상기 제4 S/P 218의 출력은 I채널 및 Q채널별로 저역여파기 219와 225를 통과하여 대역폭이 제한된 신호가 생성된다. 상기 여파기의 출력 신호는 곱셈기 220과 226에서 반송파($\cos(2\pi f_c t)$, $\sin(2$

$\pi f_{ct})$)와 곱해져서 고주파대역으로 천이되며, 합산기 227은 I채널과 Q채널의 신호를 합하여 출력한다.

<21> 상기한 도 2의 구성에 대응한 통상적인 이동국 송신기의 간략한 구성은 도 3에서 도시하고 있는 바와 같다.

<22> 상기 도3을 참조하면, 곱셈기 311, 321, 323, 325는 멀티플렉서 301, 302, 303, 304에 의해 각각 멀티플렉싱된 신호를 받아 채널구분코드(Channelization code) C_{ch1} , C_{ch2} , C_{ch3} , C_{ch43} 과 곱하기 위한 장치이다. 상기 채널구분코드(Channelization code) C_{ch1} , C_{ch2} , C_{ch3} , C_{ch43} 은 직교부호가 사용된다. 한편, 상기 멀티플렉서 301, 302, 303, 304는 각각의 채널에 채널부호화 및 인터리빙을 수행한 DPCCH 및 DPDCH 신호나 파일럿 신호, 전력제어 신호등으로 이루어지는

기타신호들이 입력되면 상기 입력되는 기타신호들과 각 TFCI 부호기로부터 제공되는 32비트의 부호화 심볼들을 멀티플렉싱하여 상기 곱셈기 211, 221, 231, 232로 각각 출력한다. 상기 32비트의 부호화 심볼들은 각 채널의 데이터 신호들에 따른 전송속도 정보를 정의하는 TFCI비트가 TFCI 부호기에 입력되어 얻어지는 심볼들이다. 채널구분코드와 곱해진 상기 곱셈기 311, 321, 323, 325의 출력은 곱셈기 312, 322, 324, 326에서 이득계수 G_1, G_2, G_3, G_4 와 곱해진다. 상기 이득계수는 G_1, G_2, G_3, G_4 는 각기 다른 값을 가질 수도 있다. 상기 곱셈기 312, 322의 출력은 제1합산기 313에서 합산되어 I채널신호로 출력되고, 곱셈기 324, 326의 출력은 제2합산기 327에서 합산되어 Q채널신호로 출력된다. 상기 Q채널신호인 제2합산기 327의 출력은 위상천이기 328에서 90도 위상이 바뀐다. 합산기 314는 제1합산기 313의 출력과 위상천이기 328의 출력을 합산하여 복소신호 $I+jQ$ 신호를 생성한다. 곱셈기 315는 상기 복소신호를 각 기지국별로 할당된 PN시퀀스(C_{scramb})에 의하여 스크램블링하며, 직병렬변환기 S/P 329에서는 상기 스크램블링된 신호를 직-병렬 변환하여 I채널 및 Q채널로 분배한다. 상기 S/P 329의 출력은 I채널 및 Q채널별로 저역여파기 316과 330을 통과하여 대역폭이 제한된 신호가 생성된다. 상기 저역여파기 316, 330의 출력 신호는 곱셈기 317과 331에서 반송파($\cos(2\pi f_c t), \sin(2\pi f_c t)$)와 곱해져서 고주파대역으로 천이되며, 합산기 318은 상기 고대역으로 천이된 I채널과 Q채널의 신호를 합하여 출력한다.

<23> 앞에서 개시한 상기 TFCI의 값은 서비스들의 조합에 따라 6비트의 기본형으

로 표현되어 1-64가지의 조합을 나타내거나, 서비스의 확장성을 위해 7-10비트의 확장형으로 표현되어 1-128, 1-256, 1-512, 1-1024개의 조합을 표현한다. 상기 TFCI의 값은 수신단에서 각 서비스들의 프레임을 해석하기 위해 반드시 필요한 정보이므로, 전송 오류가 발생한다면 수신단에서 각 서비스들의 프레임을 올바르게 해석하지 못하는 경우가 발생한다. 따라서 상기 TFCI의 값은 전송중에 오류가 발생한다할지라도 수신단에서 TFCI의 전송중에 발생하는 오류를 정정할 수 있도록 오류정정부호를 사용하여 부호화된다.

<24> 도 4에는 6비트로 표현되는 기본적인 TFCI의 오류정정부호화 방식과 부호어를 저장하고 있는 메모리의 예가 도시되어 있고, 도 5에는 7-10비트로 표현되는 확장된 TFCI의 오류정정부호화 방식과 부호어를 저장하고 있는 메모리의 예가 도시되어 있다.

<25> 상기 도 4에 도시된 바와 같이 1-64 사이의 TFCI값은 6비트로 표현되어, 상호직교(Biorthogonal) 부호기 402로 입력되어, 32심볼의 TFCI 부호어로 출력된다. 상기 도 4에서 도시하고 있는 입력정보 비트는 TFCI값의 6비트 표현형식으로 6비트가 안 되는 TFCI값은 MSB(Most significant bit, 제일 왼쪽의 bit)부터 0의 값으로 채워져 6비트 길이로 만들어진다. 상기 도 4에서는 TFCI값에 따라 메모리 혹은 기타 다른 장치에 저장되어 있는 길이 32의 부호가 선택된다. 상기 도 4의 메모리에 입력되어 있는 부호는 32 by 32 상호직교부호와 32 by 32 상호직교부호의 보수로 구성되어 있다. 상기 도 2에서 개시하고 있는 멀티플렉서 201의 LSB가 1이면 상호직교부호가 선택되고, 0이면 상호직교부호의 보수가 선택되어 TFCI 부호로 출력된다.

<26> 상기 도 4에서 도시하고 있는 상호직교부호기 402의 부호간 최단거리는 16이다. 일반적으로 2진 선형 부호(Binary Linear Codes)의 오류 정정 능력은 이진 선형부호의 각 부호간 최단 거리에 따라 결정되는데 최적부호(optimal code)가 되기 위한 2진 선형 부

호의 입력과 출력값에 따른 부호간의 최단 거리는 아래 참고문헌[1]에서 개시하고 있는 바와 같다.

- <27> **참조문헌[1] An Updated Table of Minimum-Distance Bounds for Binary Linear Codes (A.E. Brouwer and Tom Verhoeff, IEEE Transactions on information Theory, VOL 39, NO. 2, MARCH 1993)**
- <28> 상기 참조문헌[1]을 보면 (32,6) 이진 선형 부호에서 요구되는 부호간 최단 거리는 16으로, 상기 도 4에서 도시하고 있는 상호직교부호기 402는 최적코드라 할 수 있다.
- <29> 한편, 상기 도 5에는 서비스의 확장을 위하여 TFCI값이 1-128,1-256,1-512,1-1024로 확장되었을 경우 7-10비트로 표현되는 TFCI값이 입력됨에 따른 부호화방식을 도시하고 있다. 상기 도 5에서 입력정보 비트가 10비트가 안되면 제일 왼쪽 비트(MSB)부터 0으로 채워져 10비트로 표현된다. 상기 도 5의 첫 번째 도면을 참조하여 부호화 과정을 설명하면 먼저 10비트의 입력정보 비트가 디멀티플렉서 500에 입력되면 디멀티플렉서 500은 5비트 단위로 word1과 word2로 나뉘어지며, 상기 5비트 단위로 나누어진 word1과 word2는 부호화되어 각각 상호직교부호화기 502, 504에 입력된다. 상기 도 5의 두 번째 도면은 10비트로 표현되는 TFCI값을 부호화하는 알고리즘을 나타낸다. 상기 도 5에서 사용되는 부호기는 (16,5) 상호직교부호기로서 16 by 16 상호직교부호와 16 by 16 상호직교부호의 보수로 구성되어 있으며, 세 번째 도면과 같은 구조를 가진다. 상기 도 5의 알고리즘에서는 2개의 (16,5)부호기 502, 504가 사용되어 word1과 word2를 각각 부호화하여 길이 32인 부호어를 만들어 출력한다.
- <30> 상기 도 5에서 부호화된 TFCI값은 복호과정에서 다시 16비트씩 나뉘어져서 각각 5비트로 복호된다. 상기 복호과정에서 각각 복호된 word1과 word2는 합해져서 10비트의

정확한 TFCI의 값으로 된다. 이때, 상기 복호과정에서 복호된 5비트 값 중 하나라도 오류가 발생할 시에는 10비트 TFCI 전체에 오류가 발생한다. 따라서 10비트의 값이 5비트씩 나누어져 부호화된다고 할지라도 상기 부호기의 각 부호간 최단 거리는 (16,5)상호직교부호기 502, 504의 부호간 최단거리인 8이 된다.

<31> 상기 도 5에서 전송되는 TFCI의 값이 10비트이고, 부호화되는 값이 32비트임을 생각하면 참조문헌 [1]에서 요구하는 최적부호의 각 부호간의 최단거리는 12이다. 그러나 10비트의 부호어가 5비트씩 나누어져 전송되므로 인해 부호간 최단거리는 8이 되므로, 상기 도3의 오류정정방식 부호화는 최적부호를 가지지 못한다. 상기 도 5의 오류정정방식 부호화에서 최적부호를 가지지 못한다면 동일한 채널환경에서 TFCI의 오류확률이 커진다. 따라서, TFCI의 오류가 발생하여 데이터프레임의 전송률을 잘못 판단하고 데이터프레임을 복호화하면 데이터프레임의 에러률을 증가시키게 된다. 따라서, TFCI를 부호화하는 오류정정부호기는 오류율을 최소화하는 것이 중요하다.

<32> 하지만, 상술한 바와 같이 종래에는 기본적인 TFCI의 오류정정부호화 방식과 확장 TFCI의 오류정정부호화 방식에 따른 하드웨어 구성을 별도로 구비하여야 함으로서 비용과 부피에 있어 불합리한 문제점을 야기하였다. 또한, 종래 확장 TFCI의 오류정정부호화 방식은 앞서도 개시한 바와 같이 10비트의 TFCI비트를 5비트씩 나누어 처리함에 따라 최적의 부호를 가지지 못함에 따라 효율이 저하되는 문제점이 있었다.

【발명이 이루고자 하는 기술적 과제】

<33> 따라서 본 발명의 목적은 부호분할다중접속 통신시스템에서 확장된 TFCI를 부호화

할 수 있는 부호화 장치 및 방법을 제공함에 있다.

<34> 본 발명의 다른 목적은 확장된 TFCI를 부호화하는 방식이 기본적인 TFCI를 부호화하는 방식과 호환성을 가지는 부호화장치 및 방법을 제공함에 있다.

<35> 본 발명의 또 다른 목적은 부호분할다중접속 통신시스템에서 확장된 TFCI를 복호화할 수 있는 복호화 장치 및 방법을 제공함에 있다.

<36> 본 발명의 또 다른 목적은 확장된 TFCI를 복호화하는 방식이 기본적인 TFCI를 복호화하는 방식과 호환성을 가지는 복호화장치 및 방법을 제공함에 있다.

<37> 본 발명의 또 다른 목적은 부호분할다중접속 통신시스템에서 확장된 TFCI를 부호화하는 방법에 있어서 최적부호를 생성할 수 있는 부호화장치 및 방법을 제공함에 있다.

<38> 본 발명의 또 다른 목적은 부호분할다중접속 통신시스템에서 확장된 TFCI를 부호화하는 방법에 있어서 최적부호를 생성할 수 있는 방법으로 확장된 리드물러(Extended Reed Muller)를 사용하는 부호화 장치 및 방법을 제공함에 있다.

【발명의 구성 및 작용】

<39> 이하 본 발명의 일 실시 예를 첨부된 도면을 참조하여 상세히 설명하면 다음과 같다. 한편, 본 발명의 실시 예는 본 발명의 주된 내용을 구체화하기 위하여 필요한 것이며, 본 발명의 내용을 제한하지는 않는다. 또한, 본 발명의 실시 예를 설명함에 있어 앞에서 설명한 구성요소와 동일한 동작을 하는 다른 도면의 구성요소는 이전과 동일한 도면 참조번호를 사용하고 있음에 주의하여야 할 것이다.

<40> 본 발명은 앞서도 지정한 바와 같이 TFCI비트를 사용하는 CDMA 시스템에서 TFCI

를 부호화 및 복호화하는 방법에 있어서 최적부호를 생성할 수 있는 장치 및 방법으로 확장된 리드물러(Extended Reed Muller)를 CDMA시스템에 적용하는 하는 것이다.

<41> 통상적으로 선형 오류정정부호(Linear Error Correcting Code)의 성능을 나타내는 척도(measure)로서는 오류정정부호의 부호어(codeword)의 해밍 거리(Hamming distance) 분포가 있는데, 이는 각각의 부호어에서 0이 아닌 심볼의 개수를 의미한다. 즉, 0111이 어떤 부호어라면 이 부호어에 포함된 1의 개수, 즉, 해밍거리는 3이다. 이 때, 여러 부호어의 해밍거리 값들 중 가장 작은 값을 최소거리(dmin; minimum distance)라고 칭한다. 상기 선형 오류정정부호(Linear Error Correcting Code)에 있어서 상기의 최소거리가 클수록 오류정정 성능이 우수한데, 이는 참조문헌 'The Theory of Error-Correcting Codes' - F.J.Macwilliams, N.J.A. Sloane, North-Holland에서 상세히 개시하고 있다.

<42> 이때, 상기 확장된 리드물러(Extended Reed Muller)는 특정 시퀀스(sequence)와 m-시퀀스의 합으로 이루어지는 시퀀스로부터 유추해 낼 수 있는데, 상기의 시퀀스들의 합을 원소로 하는 시퀀스군을 선형 오류정정부호로 사용하기 위해서는 시퀀스군의 최소거리가 커야한다. 이러한 시퀀스군으로는 카자미 시퀀스(Kasami sequence)군, 골드 시퀀스(Gold sequence)군이나, 커독코드(Kerdock Code)군과 같은 시퀀스군들이 있다. 상기의 시퀀스들은 전체길이 $L=2^{2m}$ 일 때, 최소거리가 $(2^{2m} - 2^m) / 2$ 이고, $L=2^{2m+1}$ 일 때, 최소거리가 $2^{2m} - 2^m$ 이다. 즉, 전체길이가 32일 때, 최소거리는 12이다.

<43> 본 발명에서는 상기와 같은 시퀀스군을 사용하여 우수한 성능을 가지는 선형 오류정정부호인 확장된 오류정정부호를 생성하는 생성장치 및 방법에 대해 설명하기로 한다.

<44> 코드이론(Coding Theory)에서, m-시퀀스를 월시부호로 만드는 열치환 함수는 존재

한다. 이 때, 상기와 같은 특정 시퀀스(sequence)와 m-시퀀스의 합으로 이루어지는 시퀀스들을 상기의 열치환함수로 열치환 하게 되면, m-시퀀스 성분은 월시부호가 된다. 한편, 상기 특정 시퀀스(sequence)부분은 상기의 월시부호와의 합의 최소거리가 상기의 성질을 만족하게 되는데, 이를 이하 '마스크함수'라 칭하기로 한다. 도 6은 상기와 같은 선형 오류정정부호의 부호화 구조를 나타낸다.

<45> 하기에서는 골드 시퀀스군을 사용하여, $(2^n, n+k)$ 코드(단, $k=1, \dots, n+1$)를 생성하는 경우 상술한 마스크함수의 생성방법을 설명한다. 실제로, 골드 시퀀스는 서로 다른 특정의 m-시퀀스의 합으로 표현되어진 다는 것은 널리 알려진 사실이다. 따라서, 상기의 $(2^n, n+k)$ 코드를 생성하기 위해서는, 먼저, 길이 2^n-1 인 골드 시퀀스를 생성하여야 하는데, 생성다항식 $f_1(x)$ 과 $f_2(x)$ 로 생성되어지는 두 m-시퀀스의 합은 골드시퀀스가 된다. 이때, 각각의 m-시퀀스 $m(t)$ 는 생성다항식이 정해지면 아래에서 개시하고 있는 <수학식1>과 같이 트레이스 함수(Trace function)를 이용하여 구할 수 있다.

<46> 【수학식 1】

$$m_1(t) = \text{Tr}((A \alpha^t)), t=0, 1, \dots, 30$$

<47> 단, $\text{Tr}(a) = \sum_{k=0}^{n-1} a^{2^k}, a \in GF(2^n)$ 이다.

<48> 상기 <수학식1>에서 A는 m-시퀀스의 초기치에 따라 결정되는 값이다.

<49> 도 7은 상기의 시퀀스군중 골드 시퀀스군을 사용하여, $(2^n, n+k)$ 코드(즉, $n+k$ 비트의 정보비트를 입력하여 2^n 비트의 부호화심볼을 출력하기 위한 코드)를 생성하는 경우 상기와 같은 마스크함수의 생성과정을 나타낸다. 골드 시퀀스는 서로 다른 특정의 m-시퀀스의 합으로 표현되어진 다는 것은 널리 알려진 사실이다. 따라서, 상기의 $(2^n, n+k)$ 코드를

생성하기 위해서는, 먼저, 길이 2^n-1 인 골드 시퀀스를 생성하여야 하는데, 상기 골드 시퀀스는 앞에서도 개시한 바와 같이 생성다항식 $f_1(x)$ 과 $f_2(x)$ 로 생성되어지는 서로 다른 특정의 두 m -시퀀스의 합에 의해 생성된다.

<50> 도 7을 살펴보면, 710 단계에서는 상기의 <수학식1>에 의해 생성다항식 $f_1(x)$ 로 생성되어지는 m -시퀀스 $m_1(t)$ 와 $f_2(x)$ 로 생성되어지는 m -시퀀스 $m_2(t)$ 를 구한다. 그러면, 720단계에서는 상기의 m -시퀀스 $m_2(t)$ 를 하기의 <수학식 2>에서 나타난 월시부호로 만드는 열치환 함수 $\sigma(t)$ 를 구한다.

<51> 【수학식 2】

$$\sigma: \mathbb{W}\{0,1,2,\dots, 2^n-2\} \rightarrow \mathbb{W}\{1,2,\dots, 2^n-1\}$$

<52>
$$\sigma(t) = \sum_{i=0}^{n-1} m_2(t) 2^{n-1-i}$$

<53> 그러면, 730단계에서는 상기 m -시퀀스 $m_1(t)$ 를 0부터 30번까지 순환(cyclic shift)시켜 얻을 수 있는 31개의 시퀀스군을 상기의 $m_2(t)$ 를 월시부호로 만드는 열치환 함수 $\sigma(t)$ 의 역함수를 이용한 $\sigma^{-1}(t) + 2$ 로 열치환한 후, 각각의 시퀀스 맨 앞부분에 0을 덧붙임으로써 길이 2^n 으로 만들어서 길이 2^n 인 2^n-1 개의 시퀀스군 $d_i(t)$, $i=0,\dots,2^n-1$, $t=1,\dots,2^n$,을 생성한다. 상기과 같이 730단계에서 생성되어지는 시퀀스 군은 <수학식 3>과 같이 수식으로 표현할 수 있다.

<54> 【수학식 3】

$$\{d_i(t) | i=1,\dots,2^n, \quad t=0,\dots,2^n-2\}$$

$$\langle 55 \rangle \quad d_i(t) = \begin{pmatrix} 0, & \text{if } t = 1 \\ m_1(\sigma^{-1}(t+i)+2), & \text{if } t = 1, 2, 3, \dots, 2^n \end{pmatrix}$$

$\langle 56 \rangle$ 상기에서 구해진 $d_i(t)$ 들은 상기 마스크 함수들로 31개의 마스크로 사용할 수 있다.

$\langle 57 \rangle$ 상기에서 구해진 $d_i(t)$ 들의 성질 중 한가지 성질은 상기의 마스크들 중 두 개의 서로 다른 마스크들을 더하면 2^n-1 개 마스크 중 다른 하나의 마스크가 된다. 더 일반화시켜서 전부 0인 마스크를 포함하여 상기의 2^n-1 개의 마스크들은 2^n-1 개의 마스크 중 특정한 n 개의 마스크의 임의의 합으로 모두 표현되어질 수 있다.

$\langle 58 \rangle$ 상기 $(2^n, n+k)$ 코드를 생성할 때, 총 필요로 하는 부호어(Code word)의 개수는 모든 경우의 입력 정보비트의 가지 수인 2^{n+k} 개이다. 이 때, 길이 2^n 인 상호직교(Biorthogonal) 부호어의 개수는 $2^n \times 2 = 2^{n+1}$ 이고, 이 때, $(2^n, n+k)$ 코드를 생성하기 위해 필요한 마스크의 개수는 $(2^{n+k}/2^{n+1})-1 = 2^{k-1}-1$ 개이다. 또한, 이 때, $2^{k-1}-1$ 개의 마스크는 상기과 유사한 성질에 의해 $k-1$ 개의 마스크의 임의의 합으로 모두 표현되어질 수 있다. 따라서, 상기의 $k-1$ 개의 마스크를 고르는 방법이 필요하다.

$\langle 59 \rangle$ 상기의 $k-1$ 개의 마스크를 선택하는 방법을 설명하면, 상기 도 7의 730단계에서 $m_1(t)$ 를 0부터 2^{n-1} 번까지 순환(cyclic shift)시켜 시퀀스군을 생성하는데, 이 때, $m_1(t)$ 를 i 번 순환(cyclic shift)시킨 m -시퀀스는 상기 <수학식 1>을 사용하여 표현하면 $Tr(\alpha^i \cdot \alpha')$ 가 된다. 즉, $m_1(t)$ 를 0부터 30번까지 순환(cyclic shift)시켜 시퀀스군은 초기치 A 가 $1, \alpha, \dots, \alpha^{2^n-2}$ 에 따라서 생성되어지는 시퀀스들이다. 이 때, 갈로아체의 원소 $1, \alpha, \dots, \alpha^{2^n-2}$ 중, 선형독립인 $k-1$ 개의 원소를 찾아서 상기 $k-1$ 개의 원소를 초기치로 하는 시퀀스들을 선택하여 상기 도 7의 결과값들을 찾아 선택하면 상기과 같이 $k-1$ 개의

마스크의 임의의 합으로 $2^{k-1}-1$ 개 마스크 모두를 생성할 수 있다. 상기의 과정 중 선형 독립의 조건을 수학식으로 나타내면 하기의 <수학식4>과 같다.

<60> 【수학식 4】

$\alpha_1, \dots, \alpha_{k-1}$: 선형 독립

<61> $\Leftrightarrow c_1\alpha_1 + c_2\alpha_2 + \dots + c_{k-1}\alpha_{k-1} \neq 0, \forall c_1, c_2, \dots, c_{k-1}$

<62> 상기의 일반화된 마스크 함수생성방법을 구체적인 예를 들어 설명하기 위해서 하기에서는 골드 시퀀스군을 사용하여, (32,10)코드를 생성하는 경우 상기와 같은 마스크함수의 생성방법을 상기의 일반화된 과정에 따라 도 7을 참조하여 설명한다. 실제로, 골드 시퀀스는 서로 다른 특정의 m-시퀀스의 합으로 표현되어진 다는 것은 널리 알려진 사실이다. 따라서, 상기의 (32,10)코드를 생성하기 위해서는, 먼저, 길이 31인 골드 시퀀스를 생성하여야 하는데, 생성다항식 $x^5 + x^2 + 1$ 과 $x^5 + x^4 + x^2 + x + 1$ 로 생성되어지는 두 m-시퀀스의 합은 골드시퀀스가 된다. 또한, 각각의 m-시퀀스 $m(t)$ 는 생성다항식이 정해지면 <수학식5>와 같이 트레이스 함수(Trace function)를 이용하여 구할 수 있다.

<63> 【수학식 5】

$$m_l(t) = \text{Tr}((A \alpha^t)), t=0,1,\dots,30$$

<64> 단, $\text{Tr}(a) = \sum_{n=0}^4 a^{2^n}, a \in GF(2^5)$ 이다.

<65> 상기 <수학식5>에서 A는 m-시퀀스의 초기치에 따라 결정되는 값이다.

<66> 도 7은 상기의 시퀀스군중 골드 시퀀스군을 사용하여, (32,10)코드(즉, 10비트의 정보비트를 입력하여 32비트의 부호화심볼을 출력하기 위한 코드)를 생성하는 경우 상기

와 같은 마스크함수의 생성과정을 나타낸다. 골드 시퀀스는 서로 다른 특정의 m-시퀀스의 합으로 표현되어진 다는 것은 널리 알려진 사실이다. 따라서, 상기의 (32,10)코드를 생성하기 위해서는, 먼저, 길이 31인 골드 시퀀스를 생성하여야 하는데, 생성다항식 $x^5 + x^2 + 1$ 과 $x^5 + x^4 + x^2 + x + 1$ 로 생성되어지는 두 m-시퀀스의 합은 골드시퀀스가 된다.

<67> 도 7을 살펴보면, 710 단계에서는 상기의 <수학식1>에 의해 생성다항식 $x^5 + x^2 + 1$ 로 생성되어지는 m-시퀀스 $m_1(t)$ 와 $x^5 + x^4 + x^2 + x + 1$ 로 생성되어지는 m-시퀀스 $m_2(t)$ 를 구한다. 그러면, 720단계에서는 상기의 m-시퀀스 $m_2(t)$ 를 하기의 <수학식 6>에서 나타난 월시부호로 만드는 열치환 함수 $\sigma^{-1}(t)$ 를 구한다.

<68> 【수학식 6】

$$\sigma^{-1} : \mathbb{W}\{0,1,2,\dots, 30\} \rightarrow \mathbb{W}\{1,2,\dots, 31\}$$

<69>
$$\sigma(t) = \sum_{i=0}^4 m_2(t) 2^{4-i}$$

<70> 그러면, 730단계에서는 상기 m-시퀀스 $m_1(t)$ 를 0부터 30번까지 순환(cyclic shift)시켜 얻을 수 있는 31개의 시퀀스군을 상기의 $m_2(t)$ 를 월시부호로 만드는 열치환 함수 $\sigma^{-1}(t)$ 의 역함수를 이용한 $\sigma^{-1}(t) + 2$ 로 열치환한 후, 각각의 시퀀스 맨 앞부분에 0을 덧붙임으로써 길이 32로 만들어서 길이 32인 31개의 시퀀스군 $d_i(t)$, $i=0,\dots,31$, $t=1,\dots,32$,을 생성한다. 상기와 같이 730단계에서 생성되어지는 시퀀스군은 <수학식 7>과 같이 수식으로 표현할 수 있다.

<71> 【수학식 7】

$$\{d_i(t) | t=1,\dots,32, \quad i=0,\dots,30\}$$

$$\langle 72 \rangle \quad d_i(t) = \begin{pmatrix} 0, & \text{if } t=1 \\ m_1(\sigma^{-1}(t+i)+2), & \text{if } t=1,3,\dots,32 \end{pmatrix}$$

$\langle 73 \rangle$ 상기에서 구해진 $d_i(t)$ 들은 상기 마스크 함수들로 31개의 마스크로 사용할 수 있다.

$\langle 74 \rangle$ 상기에서 구해진 $d_i(t)$ 들의 성질 중 한가지 성질은 상기의 마스크들 중 두 개의 서로 다른 마스크들을 더하면 31개 마스크 중 다른 하나의 마스크가 된다. 더 일반화시켜서 상기의 31개의 마스크들은 31개의 마스크 중 특정한 5개의 마스크의 임의의 합으로 모두 표현되어질 수 있다.

$\langle 75 \rangle$ 상기 (32,10)코드를 생성할 때, 총 필요로 하는 부호어(Code word)의 개수는 모든 경우의 입력 정보비트의 가지 수인 $2^{10} = 1024$ 개이다. 이 때, 길이 32인 상호직교(Biorthogonal) 부호어의 개수는 $32 \times 2 = 64$ 이고, 이 때, (32,10)코드를 생성하기 위해 필요한 마스크의 개수는 $(1024/64)-1 = 15$ 개이다. 또한, 이 때, 15개의 마스크는 상기와 유사한 성질에 의해 4개의 마스크의 임의의 합으로 모두 표현되어질 수 있다. 따라서, 상기의 4개의 마스크를 고르는 방법이 필요하다. 상기의 4개의 마스크를 선택하는 방법을 설명하면, 상기 도 7의 730단계에서 $m_1(t)$ 를 0부터 30번까지 순환(cyclic shift)시켜 시퀀스군을 생성하는데, 이 때, $m_1(t)$ 를 i 번 순환(cyclic shift)시킨 m -시퀀스는 상기 <수학식1>을 사용하여 표현하면 $Tr(\alpha' \cdot \alpha')$ 가 된다. 즉, $m_1(t)$ 를 0부터 30번까지 순환(cyclic shift)시켜 시퀀스군은 초기치 A가 $1, \alpha, \dots, \alpha^{30}$ 에 따라서 생성되어지는 시퀀스들이다. 이 때, 갈로아체의 원소 $1, \alpha, \dots, \alpha^{30}$ 중, 선형독립인 4개의 원소를 찾아서 상기 4개의 원소를 초기치로 하는 시퀀스들을 선택하여 상기 도 7의 결과값들을 찾아 선택하면 상기과 같이 4개의 마스크의 임의의 합으로 15개 마스크 모두를 생성할 수 있다. 상기의 과정

중 선형독립의 조건을 수학식으로 나타내면 하기의 <수학식 8>과 같다.

<76> 【수학식 8】

$\alpha, \beta, \gamma, \delta$: 선형 독립

<77> $\Leftrightarrow c_1\alpha + c_2\beta + c_3\gamma + c_4\delta \neq 0, \forall c_1, c_2, c_3, c_4$

<78> 실제로 갈로아체GF(2⁵)에서 1, α , α^2 , α^3 은 상기와 같은 4개의 선형독립인 원소로 널리 알려진 다항식 기저(polynomial basis)이다. 따라서, 상기 다항식 기저를 상기 <수학식1>의 변수A대신 대입함으로써 구해진 4개의 마스크 함수 M1, M2, M4, M8은 다음과 같다.

<79> M1 = 00101000011000111111000001110111

<80> M2 = 000000001110011010110110111000111

<81> M4 = 00001010111110010001101100101011

<82> M8 = 00011100001101110010111101010001

<83> 하기에 나타날 실시 예에서는 상기에서 구해진 마스크를 이용한 부호기 및 복호기에 대한 실시 예를 나타낸다.

<84> 제1실시 예

<85> 상기 제1실시 예에서는 상기의 생성방법에 따른 부호화 장치 및 방법을 제공한다. 도 8은 상기 제1실시 예에 따른 부호기를 나타낸 도면이다.

<86> 도 8을 참조하면 10비트의 입력 정보비트들이 입력되면, 각각의 비트

a0, a1, a2, a3, a4, a5, a6, a7, a8, a9들은 승산기 840, 841, 842, 843, 844, 845, 846, 847, 848, 849에 입력된다. 이 때, 1비트 발생기 800에서 항상 1인 심볼들이 승산기 840에 출력되고,

한 심볼이 승산기 840으로 입력될 때마다, 승산기 840에 입력된 a0과 승산되어져 가산기 860에 입력된다. 또, 월시부호 발생기 810에서는 길이 32인 월시부호 W1,W2,W4,W8,W16이 동시에 출력되는데, 제1월시부호 W1=01010101010101010101010101 0101인 심볼들이 승산기 841에 출력되고, 한 심볼이 승산기 841로 입력될 때마다, 승산기 841에 입력된 a1과 승산되어져 가산기 860에 입력된다. 제2월시부호

W2=0011001100110011001100110011인 심볼들이 승산기 842에 출력되고, 한 심볼이 승산기 842로 입력될 때마다, 승산기 842에 입력된 a2와 승산되어져 가산기 860에 입력된다. 제4월시부호 W4=00001111000011110000111100001111인 심볼들이 승산기 843에 출력되고, 한 심볼이 승산기 843으로 입력될 때마다, 승산기 843에 입력된 a3과 승산되어져 가산기 860에 입력된다. 제8월시부호 W8=00000000111111110000000011111111인 심볼들이 승산기 844에 출력되고, 한 심볼이 승산기 844로 입력될 때마다, 승산기 844에 입력된 a4와 승산되어져 가산기 860에 입력된다. 제16월시부호

W16=00000000000000001111111111111111인 심볼들이 승산기 845에 출력되고, 한 심볼이 승산기 845로 입력될 때마다, 승산기 845에 입력된 a5와 승산되어져 가산기 860에 입력된다.

<87> 마스크 생성기 820에서는 길이 32인 마스크 함수 M1,M2,M3,M4가 동시에

출력되는데, 제1마스크 함수 M1=00101000011000111111000001110111인 심볼들이 승산기 846에 출력되고, 한 심볼이 승산기 846으로 입력될 때마다, 승산기 846에 입력된 a6과 승산되어져 가산기 860에 입력된다. 제2마스크 함수

M2=00000001110011010110110111000111인 심볼들이 승산기 847에 출력되고, 한 심볼이 승산기 847로 입력될 때마다, 승산기 847에 입력된 a7과 승산되어져 가산기 860에 입력된다.

다. 제4마스크 함수 $M4=00001010111110010001101100101011$ 인 심볼들이 승산기 848에 출력되고, 한 심볼이 승산기 848로 입력될 때마다, 승산기 848에 입력된 $a8$ 과 승산되어져 가산기 860에 입력된다. 제8마스크 함수 $M8=00011100001101110010111101010001$ 인 심볼들이 승산기 849에 출력되고, 한 심볼이 승산기 849로 입력될 때마다, 승산기 849에 입력된 $a9$ 와 승산되어져 가산기 860에 입력된다.

<88> 그러면, 가산기 860은 승산기 840,841,842,843,844,845,846,847,848,849로부터 각각 출력된 심볼들을 모두 가산하여 출력한다.


<89> 도 11은 상기 부호기의 동작에 대한 플로우 차트를 나타낸다. 도 11을 참조하면, 먼저 1100단계에서 10비트의 정보비트열 $a0, a1, \dots, a9$ 를 입력하고, 변수 sum 과 j 을 0으로 초기화한다. 그러면, 1110단계를 진행하여 변수 j 가 32인지를 판단하고, j 가 32이 아니면 1120단계를 진행하여, 수열 $W1, W2, W4, W8, W16$ 과 $M1, M2, M3, M4$ 의 j 번째 심볼인, $W1(t), W2(t), W4(t), W8(t), W16(t)$ 과 $M1(t), M2(t), M3(t), M4(t)$ 를 입력한다. 그러면, 1130단계에서는 상기 1120단계에서 입력된 심볼들을 1100단계에서 입력되어진 정보비트와 심볼단위로 곱하여 전부 합하여 sum 을 구한다. 그러면, 1140단계에서는 상기 1130단계에서 구해진 j 번째 부호화 심볼인 sum 을 출력하고, 1150으로 진행하여 j 를 1만큼 증가시킨후 다시 1110단계를 진행한다. 그러면, 다시 1110에서는 심볼의 위치를 나타내는 j 가 32인지를 판단하고 아니면 상기와 같은 동작을 반복하게 된다. 상기의 반복동작후 j 가 32가 되면 상기의 부호화 동작을 완료하게 된다.

<90> 상기 (32,10)부호기는 길이 32인 32개의 월시부호와 모든 월시부호의 심볼에 전부 1을 더한 실수의 경우 -1을 곱한, 총 64개의 부호와 4개의 마스크함수의 조합에 구해지는 총 15가지의 마스크함수의 조합으로 나타난다. 따라서, 총 부호어의 수는 1024개이다.

또한, 상기 1024개의 부호어들 중 길이 32인 32개의 월시부호와 모든 월시부호의 심볼에 전부 1을 더한 실수의 경우 -1을 곱한, 총 64개의 부호와 상기 4개의 마스크함수 중 임의의 3개의 마스크의 조합에 구해지는 총 8가지의 마스크함수의 조합으로 나타나는 (32,9)부호기, 상기 1024개의 부호어들 중 길이 32인 32개의 월시부호와 모든 월시부호의 심볼에 전부 1을 더한 실수의 경우 -1을 곱한, 총 64개의 부호와 상기 4개의 마스크함수중 임의의 2개의 마스크의 조합에 구해지는 총 8가지의 마스크함수의 조합으로 나타나는 (32,8)부호기, 상기 1024개의 부호어들 중 길이 32인 32개의 월시부호와 모든 월시부호의 심볼에 전부 1을 더한 실수의 경우 -1을 곱한, 총 64개의 부호와 상기 4개의 마스크함수중 임의의 1개의 마스크의 조합으로 나타나는 (32,7)부호기들은 모두 최소거리 12를 가진다. 상기와 같은 (32,9)부호기는 상기 도 7의 마스크함수 생성기에서 출력되는 4가지 마스크함수 중 한 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있고, (32,8)부호기는 상기 도 7의 마스크함수 생성기에서 출력되는 4가지 마스크함수 중 두 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있다. 또한, (32,7)부호기는 상기 도 7의 마스크함수 생성기에서 출력되는 4가지 마스크함수 중 세 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있다. 상기와 같은 작용을 함으로써 상기의 부호기는 입력정보 비트수에 따라 유동적으로 부호화가 가능하며, 부호기의 성능을 좌우하는 최소거리를 최대한 높임으로써 우수한 성능을 갖게 된다.

<91> 상기의 부호기는 길이 32인 월시부호들과 상기 월시부호에 모두 1을 취한 32개의 부호들, 총 64개의 부호에 상기와 같은 방법으로 구해진 15개의 길이 32인 마스크함수들을 조합한 수열들을 부호어로 사용하는데, 이에 대한 구조는 도 13에서 나타난다.

<92> 상기의 부호화과정의 이해를 돕기 위해 상기의 부호화과정에서 입력정보비트에 따



1019990027932

2000/7/2

른 모든 부호화 심볼을 나타내면 <표1a~f>과 같다.

<93>

【표 1a】

0000000000 : 00000000000000000000000000000000	0001000011 : 1000001010010010101010101010101	0010000110 : 01100111010101000010110100001
0000000001 : 11111111111111111111111111111111	0001000100 : 00011011010100001100001101000100	0010000111 : 100110000101010011101000101110
0000000010 : 01010101010101010101010101010101	0001000101 : 11100100101011110011110010111011	0010001000 : 0000110110000100110001011001000
0000000011 : 10101010101010101010101010101010	0001000110 : 0100111000000101100101000010001	0010001001 : 1111000100111011001110100110111
0000000100 : 00110011001100110011001100110011	0001000111 : 101100011111010011010011101110	0010001010 : 0101011100101110011011110011101
0000000101 : 11001100110011001100110011001100	0001001000 : 0010011101101100111111101111000	0010001011 : 10100100011010001100100001100010
0000000110 : 01100110011001100110011001100110	0001001001 : 1101100010010011000000010000111	0010001100 : 00111101111000101000011111011
0000000111 : 10011001100110011001100110011001	0001001010 : 011100100011100110101000010101	0010001101 : 110000100000111010101100000100
0000001000 : 00001111000011110000111100001111	0001001011 : 1000110111000110010101011101010	0010001110 : 011010001010010000000100101110
0000001001 : 11110000111100001111000011110000	0001001100 : 0001010001011111110011000100101	0010001111 : 100101110101101111110101010001
0000001010 : 0101101001010101010101010101010	0001001101 : 111010111010000001100110110100	0010010000 : 000000010011001001101010011000
0000001011 : 1010010101001010101010101010101	0001001110 : 0100001000010101001100100011110	0010010001 : 111111101100110110010011000011
0000001100 : 0011110000111000011110000111100	0001001111 : 101111011110101011001101100001	0010010010 : 0101010001100111001100001101101
0000001101 : 110000111000011110000111100011	0001010000 : 001010001001110000111000010001000	0010010011 : 1010101100110000110001111001010
0000001110 : 0110100101010010101001010101001	0001010001 : 1101011101100011000011110110111	0010010100 : 0011001000000001010111000001011
0000001111 : 100101101001010100101010010110	0001010010 : 011111011100100110010111011101	0010010101 : 1100110111111101010000111110100
0000010000 : 0000000001111111000000001111111	0001010011 : 100001010011001010101010010101	0010010110 : 01100111010101111010000101110
0000010001 : 11111111000000001111111100000000	0001010100 : 0001101101010111110000111011011	0010010111 : 1001000101010111110100100001
0000010010 : 0101010101010101010101010101010	0001010101 : 11100100010100000011110001000100	0010011000 : 0000111000111010110001000110111
0000010011 : 1010101001010101010101010101010	0001010110 : 010011101111010100100101011101110	0010011001 : 1111000110000101001110111001000
0000011000 : 001100111001100001100111001100	0001010111 : 10110001000001010101000100010001	0010011010 : 01010110101000001011101100010
0000011001 : 11001100001100111100110000110011	0001011000 : 001001111001001111111110000111	0010011011 : 101001001001011111001000110101
0000011010 : 01100110100110010100110010011001	0001011001 : 1101100001101100000000000111000	0010011100 : 0011110100001110010100010000100
0000011011 : 10011001100110011001100101100110	0001011010 : 0111001100011010101010101010010	0010011101 : 1100001011100001101011101111011
0000011100 : 0000111111100000000011111111111	0001011011 : 100011010011001010101010010101	0010011110 : 011000001011011000000100010001
00000110001 : 11110000000011111111000000001111	0001011100 : 0001010010100000110011001010100	0010011111 : 10010111010010011110110101110
0000011010 : 0101101010101010101010101010101	0001011101 : 11101011010111110011001101001011	0010100000 : 00000001110011011001000111000
0000011011 : 1010010101010101010101010101010	0001011110 : 0100000111101011001100111100001	0010100001 : 1111111000110010011011011000111
0000011100 : 00111100110000110011110011000011	0001011111 : 1011111000001000110011000011110	0010100010 : 010101000110001100011101101101
0000011101 : 1100001100111100110000110011100	0001100000 : 001010000100010000011110001000	0010100011 : 101010110011100011110001001010
0000011110 : 011010010100110010100110010110	0001100001 : 1101011110011110000001110111	0010100100 : 001100101111101010000100001011
0000011111 : 10001110011001100110011001101001	0001100010 : 0111110100110010101010101101101	0010100101 : 1100110100000001010111101110100
0000100000 : 0000000000000000000011111111111	0001100011 : 100000101100101010101010001010	0010100110 : 011001110101011111010000101110
0000100001 : 11111111111111110000000000000000	0001100100 : 000110101010000001111001011011	0010100111 : 10011000010101000000010110100001
0000100010 : 0101010101010101010101010101010	0001100101 : 1110010010101111100001101000100	0010101000 : 0000111011000010100111010011011
0000100011 : 1010101001010101010101010101010	0001100110 : 010011100000010101100011101110	0010101001 : 111100010011101010101010011000
0000100100 : 00110011001100111100110011001100	0001100111 : 1011000111110101001011000010001	0010101010 : 0101011100101111100100001100010
0000100101 : 11001100110011000011001100110011	0001100100 : 0010011101101100000000010000111	0010101011 : 1010010001101000001101111001101
0000100110 : 0110011001100110010100110011001	0001100101 : 1101100001000100111111110111000	0010101100 : 001111011110001101011100000100
0000100111 : 10011001100110011001100110011010	0001101010 : 01110010001110010101010110101010	0010101101 : 1100001000001110010100011111011
0000101000 : 0000111100001111111000011110000	0001101011 : 1000011011100010101010000101101	0010101110 : 0110100010100100111110101010001
0000101001 : 11110000111100000000111100001111	0001101100 : 0000101000101111001100111010100	0010101111 : 100101110101101000000100101110
0000101010 : 0101101001010101010101010101010	0001101101 : 11101011101000001100110001001011	0010110000 : 0000000100110010100100101100011
0000101011 : 10100101010010101010101010101010	0001101110 : 01000001000001000110011011100001	0010110001 : 11111110110011010101010100011000
0000101100 : 00111100001111001100001111000011	0001101111 : 10111110111010110011001000011110	0010110010 : 01010100011001111100001110001010
0000101101 : 1100001110000110011110000111100	0001110000 : 00101000100111000000111101110111	0010110011 : 1010101110011000001110001101101
0000101110 : 0110100101010011001101010101010	0001110001 : 11010111011000111111000010001000	0010110100 : 00110010000000011010000111110100
0000101111 : 100101101001100110100101010101	0001110010 : 01111101110010010101010000100010	0010110101 : 1100110111111100101111000001011
0000110000 : 00001111111000011111100000000111	0001110011 : 100000100011011010101010000101101	0010110110 : 0110100010100100111110101010001
0000110001 : 11111000000001111000011111110000	0001110100 : 0000101000101111001100111010100	0010110111 : 100101110101101000000100101110
0000110010 : 0101101001010101010101010101010	0001110101 : 11101011101000001100110001001011	0010110000 : 0000000100110010100100101100011
0000110011 : 10100101010010101010101010101010	0001110110 : 010000110111101001101001101000010001	0010110001 : 11111110110011010101010100011000
0000110100 : 00110011100110011001100000110011	0001110111 : 1011000100000010100101101110110	0010110010 : 0101011100101111100100001100010
0000110101 : 11001100001100110011001111001100	0001110100 : 0010011110010011000000001110000	0010110011 : 1010010001011100110111011000101
0000110110 : 0110100101010011001101010101010	0001110101 : 111001000101000001100110001001011	0010110100 : 0000000100110010100100101100011
0000110111 : 10011001100110011001100110011001	0001110110 : 010000010000010100110111100001	0010110001 : 11111110110011010101010100011000
0000111000 : 00001111111100001111000000000111	0001110111 : 1011111000001000110011000011110	0010110010 : 010101000110001100011101101101
0000111001 : 11111000000001110000111111110000	0001110100 : 000110100101000000011100010010100	0010110011 : 1001100001010100000001001010001
0000111010 : 0101101010100101010101010101010	0001110101 : 11101011101000001100110001001011	0010110000 : 0000000100110010100100101100011
0000111011 : 1010010101010101010101010101010	0001110110 : 010000010000010100110111100001	0010110001 : 11111110110011010101010100011000
0000111100 : 00111100001111001100001111000011	0001110111 : 1011111000001000110011000011110	0010110010 : 0101011100101111100100001100010
0000111001 : 1100001110000110011110000111100	0001110100 : 000110100101000000011100010010100	0010110011 : 1001011101011010000001001010001
0000111010 : 0101101010100101010101010101010	0001110101 : 11101011101000001100110001001011	0010110000 : 0000000100110010100100101100011
0000111011 : 1010010101010101010101010101010	0001110110 : 010000010000010100110111100001	0010110001 : 11111110110011010101010100011000
0000111100 : 00111100001111001100001111000011	0001110111 : 1011111000001000110011000011110	0010110010 : 0101011100101111100100001100010
0000111001 : 11111000000001110000111111110000	0001110100 : 000110100101000000011100010010100	0010110011 : 1001100001010100000001001010001
0000111010 : 0101101010100101010101010101010	0001110101 : 11101011101000001100110001001011	0010110000 : 0000000100110010100100101100011
0000111011 : 1010010101010101010101010101010	0001110110 : 010000010000010100110111100001	0010110001 : 11111110110011010101010100011000
0000111100 : 00111100001111001100001111000011	0001110111 : 1011111000001000110011000011110	0010110010 : 0101011100101111100100001100010
0000111001 : 1100001110000110011110000111100	0001110100 : 000110100101000000011100010010100	0010110011 : 1001100001010100000001001010001
0000111010 : 0101101010100101010101010101010	0001110101 : 111001000101000001100110001001011	0010110000 : 0000000100110010100100101100011
0000111011 : 1010010101010101010101010101010	0001110110 : 010000010000010100110111100001	0010110001 : 11111110110011010101010100011000
0000111100 : 00111100001111001100001111000011	0001110111 : 1011111000001000110011000011110	0010110010 : 0101011100101111100100001100010
0000111001 : 11111000000001110000111111110000	0001110100 : 000110100101000000011100010010100	0010110011 : 1001100001010100000001001010001
0000111010 : 0101101010100101010101010101010	0001110101 : 11101011101000001100110001001011	0010110000 : 0000000100110010100100101100011
0000111011 : 1010010101010101010101010101010	0001110110 : 010000010000010100110111100001	0010110001 : 11111110110011010101010100011000
0000111100 : 00111100001111001100001111000011	0001110111 : 1011111000001000110011000011110	0010110010 : 0101011100101111100100001100010
0000111001 : 1100001110000110011110000111100	0001110100 : 000110100101000000011100010010100	0010110011 : 1001100001010100000001001010001
0000111010 : 0101101010100101010101010101010	0001110101 : 111001000101000001100110001001011	0010110000 : 0000000100110010100100101100011
0000111011 : 1010010101010101010101010101010	0001110110 : 010000010000010100110111100001	0010110001 : 11111110110011010101010100011000
0000111100 : 00111100001111001100001111000011	0001110111 : 1011111000001000110011000011110	0010110010 : 0101011100101111100100001100010
0000111001 : 11111000000001110000111111110000	0001110100 : 000110100101000000011100010010100	0010110011 : 1001100001010100000001001010001
0000111010 : 010110101010010101010101010		

<94> 【표 1b】

0011001001	110110010101110011010101000000	0100001100	0011011011000101001100010111	0101001111	1011010000001100011110111001010
0011001010	01110001111101001100011111101010	0100001101	1100100100111010110100011101000	0101010000	0010001001100101110101110100011
0011001011	10001100000010110011100000010101	0100001110	0100001100100000110010010000010	0101010001	11011011001101000001010001011100
0011001100	000101011001001010000110001100	0100001111	10011100011011110001101011101	0101010010	011011100110001000011111011110110
0011001101	11101010011011010101111001110011	0100010000	000010100000110000110111010100	0101010011	10001000110011110100000100001001
0011001110	01000000110001111111010011011001	0100010001	111101011111001110010000010101	0101010100	0001000101010110110110001010000
0011001111	101111100111000000101100100110	0100010010	01011110100110100111010000001	0101010101	1110111010101001001001110110111
0011010000	00101001010100001100111010100111	0100010011	1010000010101100101100010111110	0101010110	010001000000011100011011000101
0011010001	110101101011100110001010110000	0100010100	0011100100110101001010001110011	0101010111	10111011111100011100000111010
0011010010	0111110000001001100100000011010	0100010101	11000110110010101011100011000	0101011000	0010110101101011100100101100
0011010011	100000111111011001101111100101	0100010110	0110110001100000011110110110010	0101011001	11010010100101010001101101001
0011010100	0001101001100010101011100111100	0100010111	10010011100111110000010010101	0101011010	0111000001111101100011111001
0011010101	111001011001110101000110000011	0100011000	0000010100001001000101001101101	0101011011	100001111100000001001100000110
0011010110	0100111100110111111011001001001	0100011001	1111101011110110110101100100100	0101011100	0001110010110011101110011111
0011010111	1011000011001000000010011010110	0100011010	010100000101100010000110001110	0101011101	1110000110100110001010000110000
0011011000	0010011001011110100100100100000	0100011011	101011110100011101111001110001	0101011110	0100101100001100100000101100100
0011011001	11011001100000101101101111111	0100011100	001101100010111010000010111	0101011111	1011010011100111110100110101
0011011010	0111001100001011100011100010101	0100011101	11001001100010111010000010111	0101100000	001000101001101000010100011100
0011011011	1000110011110100001110001101010	0100011110	011000110101111011100101011101	0101100001	1101110101100101110101101100
0011011100	000101011001101010100001011100	0100011111	10011001000001000110101000010	0101100010	011101111001111010000011110110
0011100101	11101010100100101111010001100	0100100000	0000101011110011110010011010100	0101100011	100010000110000101111000001001
0011101110	010000000111000111010000100110	0100100001	1111010100000110000110110010101	0101100100	000100011010100100100111100000
0011101111	10111111000111000010111011001	0100100010	010111110101100101100011000001	0101100101	111011100101011011010000110111
0011100000	001010010101110011000100100111	0100100011	101000000101011010011100111110	0101100110	010001001111100011100111000101
0011100001	110110010101110110110110110000	0100100100	0011100111001010101111100111	0101100111	101110110000001100011010011010
0011100010	0111110011110110011011100011010	0100100101	1100011000110101001010000011000	0101101000	0010110100101010001101101100
0011100011	10000011000000100100001100101	0100100110	011010000101111000001011001010	0101101001	1101001001101011101001000101
0011100100	000110101001110101000010111100	0100100111	10010011010000001111010100101	0101101010	0111000011000000010011101111001
0011100101	11100101010001010111010000011	0100101000	00000101111010111010111011011	0101101011	1000011100111111011000100000110
0011100110	0100111110010000000010000101001	0100101001	11111010000010010001010000100100	0101101100	000111010100110001000001001111
0011100111	101100000110111111101111010110	0100101010	0101000010100011101111010001110	0101101101	11100001010100110101101100000
0011101000	001001101010000101101010100000	0100101011	1010111101011100010000010110001	0101101110	010010111110011011110111001010
0011101001	110110010101110100100101011111	0100101100	0011011000101110110001101000	0101101111	10110100000011001000001010101
0011101010	01110011111100001110000010101	0100101101	110010010011101000101100010111	0101110000	001000100110010100010100011100
0011101011	10001100000101110001111101010	0100101110	011000111001000010001101111101	0101110001	11011101100110101110101100011
0011101100	0011010110010010010111100111001	0100101111	1001110001011110111001001000010	0101110010	01110111001100000001000010001
0011101101	11101010011011011010000110001100	0100101000	000010100000010111001000010101	0101110011	100010001100111101111011110110
0011101110	01000000110001110000101100100110	0100101001	111101011111001000110111010100	0101110100	0001000101010110001001101111
0011101111	1011111001110001111001011001	0100101010	01011110100101110100010111110	0101110101	11101110101010110110001010000
0011110000	0010100101000010110001010110000	0100101011	101000001010100010011101000001	0101111010	010001000000011011100000111010
0011110001	1101010101011101001110101001111	0100101100	001110010011010111011100011000	0101111011	101110111111100100011011000101
0011110010	011111000000100001101111100101	0100101101	11000110100101001000110001110011	0101111000	001011010110101000011011001011
0011110011	100000111111011110010000011010	0100101110	011011000110000010000100100101	0101111001	110100101001010111001001010100
0011110100	000110100110001001000110000011	0100101111	100100111001111011110110110010	0101111010	011100000111111010011100000110
0011110101	111001011001101101011100111100	0100111000	00000101000001110101100100100	0101111011	1000011111000000010100011111001
0011110110	01001110011011000000011010110	0100111001	111110101111010000101001101011	0101111100	000111000101001001010000110000
0011110111	10110000110010001111011001001	0100111010	010100000101100101111001110001	0101111101	111000011010011011010111001111
0011111000	001001100101110011011011011111	0100111011	101011110100011010000110001110	0101111110	0100101100001100011111010010101
0011111001	1101100110100001100100100100000	0100111100	00110110001110101101000001011	0101111111	1011010011110011100000101001010
0011111010	0111001100001010011100011101010	0100111101	1100100111000101001001111101000	0110000000	0000101100110000011101011101100
0011111011	100011001111010011000111000101	0100111110	011000110111110001101010000010	0110000001	11110100110010111000100100010011
0011111100	0001010101110010001101100101011	0100111111	100111001000000011100101011101	0110000010	0101111001100001001000111011001
0011111101	1110101010010101010000101110011	0101000000	0010001010010101110101101011100	0110000011	1010000110011110101110001000110
0011111110	010000000011000000010111011001	0101000001	1101110101100101000101001000011	0110000100	0011100000000111010001011101111
0011111111	1011111111000111111010000100110	0101000010	01101111000111101111000001001	0110000101	1100011111110001011101000100000
0100000000	0000101011110010001101100101011	0101000011	1000100000110000010000011110110	0110000110	01101101010100100001000010001010
0100000001	11110101000001101110010011010100	01010000100	000100011010100111010000110111	0110000111	100100101010110111011110110101
0100000010	0101111110101100010011100111110	01010000101	1110111001010110001001111001000	0110000100	000001000011101011110011100011
0100000011	101000001010011101000110000001	01010000110	0100010011111001000110100111010	0110000101	11111011110001001000011000011100
0100000100	0011100111001010100000011000	01010000111	101110110000000110110001010001	0110000100	010100010101110001011000101010
0100000101	1100110001010111010111100111	01010000100	00101101001010111001000100011	0110000101	1010110100000111001011001001
0100000110	011011001001111011110101001101	01010000101	110100011010100001101110110100	0110000100	00110110000100000100101010000
0100000111	1001001101100001000001010110010	0101000100	011100011000000101000100000110	0110000101	110010001111011101101010010111
0100001000	0000010111110100001010000100100	0101000101	100001110011111010011101111001	0110000110	010000010111010001111110000101
0100001001	111110100000100111101011110111	01010001100	00011110100110111010000110111	0110000111	10011011010001011100000111010
0100001010	011110100000100111101011110111	01010001101	111000010101000001101110110100	0110000100	00011110000100010010101010000
0100001011	111110100000100111101011110111	01010001101	111000010101000001101110110100	0110000101	111000011010011011010111001111
0100001100	0101000001010000010101100110010	01010001101	111000010101000001101110110100	0110000110	010000010111010001111110000101
0100001101	111110100000100111101011110111	01010001101	111000010101000001101110110100	0110000111	10011011010001011100000111010
0100001110	0101000001010000010101100110010	01010001101	111000010101000001101110110100	0110000100	000010111100010110111010000101
0100001111	111110100000100111101011110111	01010001101	111000010101000001101110110100	0110000101	1111100001100010110110110001011
0100001110	0101000001010000010101100110010	01010001101	111000010101000001101110110100	0110000100	000010111100010110111010000101
0100001111	111110100000100111101011110111	01010001101	111000010101000001101110110100	0110000101	1111100001100010110110110001011
0100001110	0101000001010000010101100110010	01010001101	111000010101000001101110110100	0110000100	000010111100010110111010000101
0100001111	111110100000100111101011110111	01010001101	111000010101000001101110110100	0110000101	1111100001100010110110110001011
0100001110	0101000001010000010101100110010	01010001101	111000010101000001101110110100	0110000100	000010111100010110111010000101
0100001111	111110100000100111101011110111	01010001101	111000010101000001101110110100	0110000101	1111100001100010110110110001011
0100001110	0101000001010000010101100110010	01010001101	111000010101000001101110110100	0110000100	000010111100010110111010000101
0100001111	111110100000100111101011110111	01010001101	11100001010100000110		

<95> 【ㄩ 1c】

0110010010	0101111010011110001000101000110	0111010101	1110111011001000100101010101000	1000011000	00010011110001110010000010100001
0110010011	10100000101000011101110010111001	0111010110	0100001111001101110000000000010	1000011001	111011000011100010111110111110
0110010010	0001110001111000010001001000000	0111010111	011101000110000100011111111101	1000011010	01000110100001000101101011101010
0101010101	11000111000001110111010101011111	0110101000	000101001001011110001001010101	1000011011	1011010010101011000101000001011
0110010110	01101011010101000100000111010101	0111011001	110100111010000111010100100100	1000011100	001000001110100000001110010010
0110010111	10010010001010010111011111001010	011101010	011110011111010101100000011110	1000011101	11011110000101111010000101101
0110011000	00000100110001000111100100011100	0111010111	100001100000101010101111000001	1000011110	0111010101000001000010110000111
0110011001	11111011001101110000101011100011	0110101000	000111110101001011101001011000	1000011111	1000101000111101011100100111000
0110011010	0110001100100010010110001001001	0111011101	1110000001101010100001010100111	1000100000	000111000011011110100001010110
0110011011	1010110011011101101001010010110	0111011110	010101010100000111011100001001	1000010001	11100001110000000001011010001
0110011100	0010111111010101001000100010110	0111010111	101010101111100001000001110010	1000010010	01010010110001010000000111101
0110011101	110010000001000101010111010000	0111000000	001000110101011101110010100100	1000100011	101010100011101011101000000101
0110011110	01100001010001000001110111110101	0111000010	110110101010000100001010010101	1000010010	001011110000010001100001110100
0110011111	10011101010101111000100001000101	0111000011	0111100000001000101000010001	1000010101	1101000001111010001110001100010
0110100000	00001010100101001000100100010011	0111000011	1000010001111110110101111001110	1000010110	01110100100101000010101100010
0110100001	1111010011001011011011110110	0111100100	000100000110010001010001010111	1000010111	10000101010111000100010011011
0110100010	0101111011000001120011110000110	0111100101	010101010100001110101010101000	1000010100	0001001100010001000001000001000
0110100011	101000010011010000010111010011	0111100110	01010100011110000000000000110	1000010001	1110001100010001000001000001000
0110100100	001011100000000110111000011000	0111101111	101010111000000111011111100010	1000010010	001011110000011010000101011100
0110100101	1100011111100001000101101111	0111101100	011010101010000100000100010011	1000010011	11010000011100001101000010101
0110100110	0110101010100101011101111010101	0111101101	10000100000001010010100110001	1000011110	0111010010110101011100011000111
0110100111	1001001010101000010000100001010	0111110000	001000000100101010101010101000	1000011010	000100000111000001101000010101
0110101000	00000100001110111000011000011100	0111110001	011110000100111100001001100100	1000011011	1000000101000001000001000001000
0110101001	1111101111000000011200111100011	0111110010	011010101111101001010011001110	1000011000	0000011000111101111010000010101
0110101010	01010001011011010100101001001001	0111110011	11100000010010101010101010111	1000010001	1111000001010000000111101010010
0110101011	10101110100010010100101001010110	0111110010	01010100011110000000000000110	1000010010	0101001010100001000001001110011
0110101100	0010111000010001010100101001011	0111110101	101010111000000111011100001001	1000010011	1001101111100001101010100001000
0110101101	01100001010000101100000100001010	0111110110	010101010100000111010100001000	1000010010	0000011000111101111010000010101
0110101110	10011000000100000111010100001010	0111110111	101010101010000100000100010010	1000010011	1110000001010000000111101010010
0110101111	1001110101010100000111010100100	0111000000	000111100000000000011101000010	1000010010	0101001010101001000101011111
0110100000	0010001010101111000010010011011	0111000001	110100000101011101110100001010	1000010011	1000010101010100010000010111000
0111000001	1101110010101000011110010100100	0111000010	011010000111011110111000101101	1000010010	0000011000111101111010000010101
0111000010	01101100000001010100111100110	0111000011	100000101010110101010101010000	1000010011	10000001010101101010101000000101
0111000011	100001001010101010101010101000	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010101010000	1000010011	10000001010101101010101000000101
0111000011	111111011100110101010101010101	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101100110100100101010111	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	0110010101000111000001111010101	0111000011	100000100100110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101001100000111100000010	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	1110010101000010110000010000101	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101100110100100101010111	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	0110010101000111000001111010101	0111000011	100000100100110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101001100000111100000010	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	1110010101000010110000010000101	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101100110100100101010111	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	0110010101000111000001111010101	0111000011	100000100100110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101001100000111100000010	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	1110010101000010110000010000101	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101100110100100101010111	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	0110010101000111000001111010101	0111000011	100000100100110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101001100000111100000010	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	1110010101000010110000010000101	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101100110100100101010111	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	0110010101000111000001111010101	0111000011	100000100100110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101001100000111100000010	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	1110010101000010110000010000101	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101100110100100101010111	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	0110010101000111000001111010101	0111000011	100000100100110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101001100000111100000010	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	1110010101000010110000010000101	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101100110100100101010111	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	0110010101000111000001111010101	0111000011	100000100100110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101001100000111100000010	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	1110010101000010110000010000101	0111000010	011010101010000101101000000101	1000010010	0000011000111101111010000010101
0111000010	000100000110010010101010101000	0111000011	100000101010110101010100000101	1000010011	10000001010101101010101000000101
0111000011	101110101100110100100101010111	0111000010	01101010101000010		

<96> 【표 1d】

1001011011 : 10010001000011001110100111100	1010011110 : 01101000110110000101010000000	1011100001 : 1100101001100110101100101100001
1001011100 : 000010001001011111000111100101	1010011111 : 1000101110010011101010011111111	1011100010 : 01100000110011000001100001001011
1001011101 : 11110111011010000001110000011010	1010100000 : 000110111111010111010101010101	1011100011 : 1001111100110011110011101010100
1001011110 : 010110111000010101101010110000	1010100001 : 11100010000001010100001010010110	1011100100 : 00000110101010100111110000101101
1001011111 : 10100010001111010100101001111	1010100010 : 0100100010101111110100000111100	1011100101 : 111100101010101100000111010010
1001100000 : 00110100010101000010000011011001	1010100011 : 101101101010000000101111000011	1011100110 : 010100111111111001010101111000
1001100001 : 110010111010101111011100100110	1010100100 : 001011101101001100011100101010	1011100111 : 10101100000000001101010010000111
1001100010 : 0110000100000001011010110001100	1010100101 : 1101000100110100111000110010101	1011101000 : 00111010100101100100001000010001
1001100011 : 100111011111010000101001110011	1010100110 : 0111101100111001101101100001111	1011101001 : 110001010110100110111011101110
1001100100 : 000001101100111000100111101010	1010100111 : 10000100011000110010010011110000	1011101010 : 0110111110000110001011101000100
1001100101 : 11111000100110001110110000010101	1010101000 : 00010010111101011011001001100110	1011101011 : 100100000111100111010001011011
1001100110 : 010100100011001000011010111111	1010101001 : 11101101000010100100110110011001	1011101100 : 000010011010010111000100100010
1001100111 : 10101101110011010111001000000	1010101010 : 01000111101000001110011001100111	1011101101 : 111101100101101000011101101101
1001101000 : 001110101011011001011111010110	1010101011 : 10111000010111110001100011001100	1011101110 : 010110011110000000100001110111
1001101001 : 11000100100100100110100000010001	1010101100 : 00100001110001101000000101010101	1011101111 : 101000110000111110101110010000
1001101010 : 0110111000001110011110101000011	1010101101 : 1101111000111001011111010101010	1011110000 : 0011010101100110010010111100001
1001101011 : 1001000111110001100001010111100	1010101110 : 011101010010011101010000000000	1011110001 : 110010101001100110010000011110
1001101100 : 00001000011010000001110011100101	1010101111 : 100010110110110000101011111111	1011110010 : 0110000000110010001100001010101
1001101101 : 1111011110010111110001100011010	1010110000 : 0001110100000101011110110010110	1011110011 : 1001111100011001110011101001011
1001101110 : 0101110100111010100100110110000	1010110001 : 1110000111110101000000101101001	1011110100 : 00000110010101011111011010010
1001101111 : 10100010110000101011010001111	1010110010 : 01001000010100001110100011000011	1011110101 : 111100110101010100000010010101
1001110000 : 0011010001010101001000000100110	1010110011 : 1011011101011110001011100111100	1011110110 : 0101001100000000001010110000111
1001110001 : 110010110101010011011111011001	1010111000 : 0010111000110110000111010100101	1011110111 : 10101100011111111101010001111000
1001110010 : 0110000111111100111010101110011	1010111001 : 1101000111001001011100010101010	1011111000 : 001101001101001010000101101110
1001110011 : 1001111000000001100010100001100	1010111010 : 01110101100001110101111110000	1011111001 : 110001010010101011101000010001
1001110100 : 00000111100110000001001100010101	1010111011 : 1000010010011100001001000000111	1011111010 : 0110111100111100000011101101101
1001110101 : 111100001100111110110011101010	1010111100 : 0001001000001010110010010011001	1011111011 : 1001000011000011111000010000100
1001110110 : 0101001010011010100001100100000	1010111101 : 1110110111101010100110101100110	1011111100 : 00001001010101011100011011011
1001110111 : 10101101001100101110011011111	1010111010 : 0100011101011111110011110011100	1011111101 : 1111011010010110000110000100010
1001111000 : 0011101101001000010111100101001	1010111101 : 1011100010100000001100000110011	1011111110 : 010111000000111100100100100001000
1001111001 : 1100010001011011101000011010110	1010111110 : 00100001001110011000000110101010	1011111111 : 1010001111100001101101101110111
1001111010 : 0110111011100010111101001111100	1010111101 : 1101111011000110011111001010101	1100000000 : 00010110011100011000011101010
1001111011 : 1001000100001110100001011000011	1010111110 : 0111010001101100110101001111111	1100000001 : 11101001001100011100101100001011
1001111100 : 0000100010010111000110000011010	1010111111 : 10001011001001100101010000000	1100000010 : 01000011100110110110000100101111
1001111101 : 1111011101100001110001111100101	1011000000 : 0011010110011001101100101100001	1100000011 : 10111100011001001001111011010000
1001111110 : 0101110111000010010010100101111	1011000001 : 11001010011001100100110100011110	1100000100 : 0010010111111010000011101001001
1001111111 : 100001011011011001101010110000	1011000010 : 01100000011001100111100111010100	1100000101 : 110110100000001011110001001010
1010000000 : 0001110111110100100001010010110	1011000011 : 1001111001100110001100001001011	1100000110 : 01110000101010000101001000011100
1010000001 : 1110001000000010111101010101001	1011000010 : 00000110101010101000000111010010	1100000111 : 1000111101010111101011011100011
1010000010 : 010010001010111000101111000011	1011000011 : 111100101010101011111000010101	1100001000 : 00011001110000010011101110101
1010000011 : 1011011101010000111010000011100	1011000010 : 01010011111111111101010010000111	1100001001 : 11100110001111110110000100001010
1010000100 : 00101110110010010111000110100101	1011000011 : 1010110000000000010101111000	1100001010 : 01001100100101000110111000100000
1010000101 : 1101000100110110000111001011010	1011000000 : 001110101001010111110111101110	1100001011 : 101100110101011100100011101111
1010000110 : 0111101110011100010001011110000	1011000001 : 11000101011010010100001000010001	1100001100 : 0010101011110010000000000001010
1010000111 : 1000010001100011110110100001111	1011000010 : 011011111000011111010001011011	1100001101 : 11010101000011011110111011001
1010001000 : 000100101111010100110110011001	1011000011 : 100100000011100000101110000100	1100001110 : 0111111101001110101110100010011
1010001001 : 111011010000101011001001100110	1011000100 : 00001001100101011000011101101101	1100001111 : 100000000110000101000101111000
1010001010 : 01000011100010011111010101010	1011000101 : 1111011001011000111000100100010	1100010000 : 00010110001100010011010010000101
1010001011 : 10111000011100110000001010101	1011000110 : 0101110011110000110111110001000	1100010001 : 111000111001110110010110111010
1010001100 : 00100001110001100111110101010	1011000111 : 1010001000011110010010000111011	1100010010 : 01000011010010000110000111010000
1010001101 : 110111100011100110000001010101	1011000110 : 00110101011001101011001000001110	1100010011 : 10111100011011001111000101111
1010001110 : 01110100100110010101111111111	1011000111 : 110010101001100100101111100001	1100010100 : 00100101000000100000011110110110
1010001111 : 1000010110110110011010000000000	1011000100 : 0110000000110011110011101001011	1100010101 : 110101011111101111100001001001
1010010000 : 00011101000001010100001001101001	1011000101 : 1001111110011000001100010110100	1100010110 : 01110000001011010101011100011
1010010001 : 0100010111110101011110110010110	1011000100 : 0000011001010101100000010010101	1100010111 : 10001111101010000101011000011100
1010010010 : 1110101110101111110100011000011	1011000101 : 111100101010101011111011010010	1100011000 : 0001100100111100001110110001010
1010010011 : 1011011110101111110100011000011	1011000110 : 1010110011111110010101110000111	1100011001 : 111001101100000111000001110101
1010010100 : 0010110001101100110000101011010	1011000111 : 110010011111110010101110000111	1100011010 : 010011000110101101101101101111
1010010101 : 11010001110010011000111010100101	1011000100 : 0011101001100101111101000010001	1100011011 : 101100111001010010010000100000
1010010110 : 0111011011000110010000000001111	1011000101 : 11000101011010110010000010110110	1100011100 : 00101010000011010000100001011001
1010010111 : 1000010010011100110111110000	1011000100 : 011011111000011111010001011011	1100011101 : 11010101111001011110110001010
1010011000 : 0001001000001010011010110011001	1011000101 : 100100001100001011100001011101011	1100011110 : 011111110101100000101111110110
1010011001 : 111011010000101011001001100110	1011000110 : 000010010101010000111000000010110	1100011111 : 100000000100000101000101111000
1010011010 : 0100011101011110000000011001100	1011000111 : 1111011001011000111000100100010	1100010000 : 00010110001100010011010010000101
1010011011 : 10111000011100110000001010101	1011000100 : 010111001111000011011110001000	1100010001 : 111000111001110110010110111010
1010011100 : 00100001110001100111110101010	1011000101 : 1001000000000000000000000000000	1100010010 : 0100001101001000011000000000000
1010011101 : 110111100011100110000001010101	1011000100 : 0001001000000000000000000000000	1100010011 : 1110000000000000000000000000000
1010011110 : 01110100100110010101111111111	1011000101 : 111100101010101011111101100010	1100011000 : 0001100100111100001110110001010
1010011111 : 1000010110110110011010000000000	1011000110 : 1010110011111110010101110000111	1100011001 : 111001101100000111000001110101
1010011000 : 00011101000001010100001001101001	1011000111 : 1100100111001110011101000010000	1100011010 : 010011000110101101101101101111
1010011001 : 0100010111110101011110110010110	1011000100 : 0001001001111000011101100001000	1100011011 : 101100111001010010010000100000
1010011010 : 1110101111010110110010010001110	1011000101 : 11000101011010110010000010110110	1100011100 : 00101010000011010000100001011001
1010011011 : 1000010001100011110110100001111	1011000100 : 0110111110000111110100001011011	1100011101 : 11010101111001011110110001010
1010011000 : 000100101111010100110110011001	1011000101 : 100100000011100000101110000100	1100011110 : 011111110101100000101111110110
1010011001 : 111011010000101011001001100110	1011000110 : 00001001100101011000011101101101	1100011111 : 100000000100000101000101111000
1010011010 : 0100011101011110000000011001100	1011000111 : 1111011001011000111000100100010	1100010000 : 00010110001100010011010010000101
1010011011 : 10111000011100110000001010101	1011000100 : 0101110011110000110111110001000	1100010001 : 111000111001110110010110111010
1010011100 : 00100001110001100111110101010	1011000101 : 1001000000000000000000000000000	1100010010 : 0100001101001000011000000000000
1010011101 : 110111100011100110000001010101	1011000100 : 0001001000000000000000000000000	1100010011 : 1110000000000000000000000000000
1010011110 : 01110100100110010101111111111	1011000101 : 111100101010101011111101100010	1100011000 : 0001100100111100001110110001010
1010011111 : 1000010110110110011010000000000	1011000110 : 1010110011111110010101110000111	1100011001 : 111001101100000111000001110101
1010011000 : 00011101000001010100001001101001	1011000111 : 1100100111001110011101000010000	1100011010 : 010011000110101101101101101111
1010011001 : 0100010111110101011110110010110	1011000100 : 0001001001111000011101	

<97> 【표 1e】

1100100100 : 001001011111101111100010110110	1101100111 : 101001110011010010001001101011	1110101010 : 0100110101011001111110000011000
1100100101 : 11011010000000100000011101001001	1101101000 : 0011000110100010001101001111101	1110101011 : 1011001010100110000000111100111
1100100110 : 0111000010101000101011011100011	1101101001 : 110011001011011100101100000010	1110101100 : 001010110011111100110100111110
1100100111 : 1000111010101110101001000011100	1101101010 : 0110010011110111010000110101000	1110101101 : 1101010011000000011001010000001
1100101000 : 00011001110000011100010010001010	1101101011 : 100110110000100010011100101011	1110101110 : 0111111001101010110011110010101
1100101001 : 1110011000111110001110101110101	1101101100 : 00000010100100010000011111001110	1110101111 : 10000001100101010011000011010100
1100101010 : 0100110010010100100100001101111	1101101101 : 111111010101101111100000110001	1110110000 : 000101111111100101001101011101
1100101011 : 10110011010101010111000100000	1101101110 : 010101111000100010100101011011	1110110001 : 1110100000000110101100101000010
1100101100 : 00101010111001011101110111011001	1101101111 : 10101000001110110101010101100100	1110110010 : 01000010101001111100111101000
1100101101 : 11010101000011010000100001000110	1101110000 : 00111110010100100011101100001101	1110110011 : 1011101010101100000110000010111
1100101110 : 0111111101001111010001011101100	1101110001 : 110000010101011100010011110010	1110110100 : 001001001100111100101010001110
1100101111 : 1000000010110000101110100010011	1101110010 : 0110101100000111010111001011000	1110110101 : 1101101100110000011010001100001
1100110000 : 0001011000110001110010110111010	1101110011 : 1001010011111000100100011010011	1110110110 : 011100011001010110000001101011
1100110001 : 11101001110011100011010010000101	1101110100 : 000011010110000100001000000111110	1110110111 : 10001110011001010011111100100100
1100110010 : 0100001011001001001111000101111	1101110101 : 111100101001110111011101111000001	1110111000 : 00010000110011110010101001010010
1100110011 : 1011110010011010110000111010000	1101110110 : 0101100000110100010111010110101	1110111001 : 111001110000110001010100100101
1100110100 : 001010100000001111100001001001	1101110111 : 1010001110010111010001010010100	1110111010 : 010011011000110111110011100111
1100110101 : 110110101111010000011110110110	1101111000 : 0011000101111001010011000000010	1110111011 : 10110010010111001111101010100
1100110110 : 0111000010101111010110100011100	1101111001 : 110011101010001011001011111101	1110111100 : 001010111000000100110101000001
1100110111 : 100011110101000010100101100011	1101111010 : 01100100000100001000010101011	1110111101 : 11010100001111110110010111110
1100111000 : 0001100100111101100010001110101	1101111011 : 1001101111101111001111010101000	1110111110 : 01111101001011100111110101000
1100111001 : 1110011011000001001110110001010	1101111100 : 00000010011011100000011100110001	1110111111 : 10000001011010001100000101011
1100111010 : 0100110001101011100100010010000	1101111101 : 11111101100100001111100011001110	1111000000 : 001111110110000010100111001010
1100111100 : 0010101000001101111011101000110	1101111110 : 0101011100111011010101001001000	1111000001 : 110000010011110101011000110101
1100111101 : 110101011110010000100010111001	1101111111 : 1010100011000100101011011001011	1111000010 : 01101010001101011111100010011111
1100111110 : 0111111010110001010001000010011	1101111111 : 10100000001101011001101111011	1111000011 : 1001010111001100000001101100000
1100111111 : 100000001010111010111011101100	1101111111 : 1111101101001001111100011001110	1111000100 : 00001100010001100110010101111001
1101000000 : 0011111010101101110001000001101	1101111111 : 10100010010101110001100110010	1111000101 : 1111100111001111010011011001001
1101000001 : 110000010100100011101111110010	1101111111 : 10100010010101110000010010000	1111000110 : 0101010000100111110000010100011
1101000010 : 011010111111000010010000101011000	1101111111 : 10100000010001010110101010010	1111000111 : 10100111110110011111010111001
1101000011 : 1001010000000111011011010100111	1101111111 : 11000111010110110000000100100	1111000111 : 10100111110110011111010111001
1101000100 : 0000110110001110111011100111110	1101111111 : 10001000000100010101101010010	1111000111 : 100110101000110000010001101111
1101000101 : 111100101000100001000010000111110	1101111111 : 1100011111100111010100101001101	1111000111 : 10100111110110011111010111001
1101000110 : 0101100011001011010001001101011	1101111111 : 10100110101011000000111100111	1111000111 : 10100111110110011111010111001
1101000111 : 1010011110011010001011101100100	1101111111 : 10100010101011000000111100111	1111000111 : 11111100101000110110101000001001
1101001000 : 0011000101001010010110000001000	1101111111 : 101100101010111110000011000	1111000111 : 0101011000001011111000001100011
1101001001 : 1100111001011010011010011111101	1101111111 : 10010101001111101100110000001	1111000111 : 1010100111110110001111110101100
1101001010 : 0110010111110001111111001010111	1101111111 : 101010110000001001101001111110	1111000111 : 111010011110101110100100110101
1101001011 : 10011011000010000110000110101000	1101111111 : 011111001100100011000110000110100	1111000111 : 11000000110000001010110101010
1101001100 : 0000001010010001111100000110001	1101111111 : 10001110011111100001111001110	1111000111 : 011010101100101111110001100000
1101001101 : 1111110101011100000011111001110	1101111111 : 11010000000001101001101011101	1111000111 : 10010101001101010000001110011111
1101001110 : 01010111100010010101010100100	1101111111 : 10100010101011000000111100111	1111000111 : 000001101001110010101000001001
1101001111 : 1010100000110101010010100101011	1101111111 : 01000010101011011100001010011	1111000111 : 111111001011000110101011110110
1101010000 : 00111110010100101100010011110010	1101111111 : 10111010101011111001111101000	1111000111 : 01010110000011001111010100101
1101010001 : 11000001101011010011101100001101	1101111111 : 10001001100111101101001110001	1111000111 : 101001100000110001100001010100
1101010010 : 0110101100000111001000110100111	1101111111 : 11011011001100001001010110001110	1111000111 : 0011000010010000101001100011010
1101010011 : 100101001111000011011001011000	1101111111 : 011100011001110100011111001000	1111000111 : 1100111010111101011100111000101
1101010100 : 000011010110000111101111000001	1101111111 : 1000111001100101110000011011011	1111000111 : 10011010100011101000001100100000
1101010101 : 11110010100111100000100000111110	1101111111 : 1100011111100111010100101001101	1111000111 : 0000001101000111001010100001001
1101010110 : 01011000001101001010001010010100	1101111111 : 1010011011000110000000110001000	1111000111 : 11111100010110000110101011110110
1101010111 : 1010011111000101010111010101011	1101111111 : 101100100101001111110001100111	1111000111 : 01010110111011011000000011100
1101011000 : 001100010101110110010111111101	1101111111 : 101100100101001111110001100111	1111000111 : 1010100100001001001111110100011
1101011001 : 11001110101000100011010000000010	1101111111 : 10101010000111111001101010000001	1111100000 : 0011111011000000101011000110101
1101011010 : 01100100000010001001111010101000	1101111111 : 0111110100101001001100000010101	1111100001 : 110000001001111010000011001010
1101011011 : 1001101111101011011000010101011	1101111111 : 10110110100101001100000010101	1111100010 : 01101010001101010000001101100000
1101011100 : 000000100110111110000110001	1101111111 : 10000001010101011001111101010	1111100011 : 100101011100000011111000111111
1101011101 : 11111101100100010000011100110001	1101111111 : 11010000000001101001101011101	1111100100 : 00001100010100101101000000110
1101011110 : 01010111001101110101011001101	1101111111 : 01000010101011011100001010111	1111100101 : 111100110100110110101000001001
1101011111 : 10101000110001000101001100100	1101111111 : 10111010101011111001111101000	1111100110 : 0101001111110001100111101010011
1101100000 : 0011111010101001110111110010	1101111111 : 10111010100100000100111010000	1111100111 : 101001101111001110011110101100
1101100001 : 110000010101001011000000001101	1101111111 : 101010000110000100101010110001	1111100111 : 101001101111001110011110101100
1101100010 : 0110101111100001101101010001110	1101111111 : 1101011110011110110101010001110	1111100111 : 00110000011011101010100000110
1101100011 : 10101000110001000101001100100	1101111111 : 010000101010110111001100010111	1111100101 : 111100111010110010010101111001
1101100011 : 0011111010101001110111110010	1101111111 : 10111010100100000100111010000	1111100110 : 01011001000001100011000001010011
1101100011 : 110000010101001011000000001101	1101111111 : 101010000110000100101010110001	1111100111 : 101001101111001110011110101100
1101100011 : 01101011111000011011101010001110	1101111111 : 1101011110011110110101010001110	1111100111 : 00110000011011101010100000110
1101100011 : 100101000000111100000101011000	1101111111 : 01110001011001011000000001000	1111100111 : 1100111101011001001010100000110
1101100100 : 00001101100111100001000110000001	1101111111 : 1000111001100101110000001101101	1111100111 : 10011010100011101000001100100000
1101100101 : 11110010100111100000100000111110	1101111111 : 1100011111100111010100101001101	1111100111 : 0000001101000111001010100001001
1101100110 : 01011000001101001010001010010100	1101111111 : 1010011011000110000000110001000	1111100111 : 11111100010110000110101011110110
1101100111 : 1010011111000101010111010101011	1101111111 : 101100100101001111110001100111	1111100111 : 01010110111011011000000011100
1101101000 : 001100010101110110010111111101	1101111111 : 101100100101001111110001100111	1111100111 : 1010100100001001001111110100011
1101101001 : 11001110101000100011010000000010	1101111111 : 10101010000111111001101010000001	1111100000 : 0011111011000000101011000110101
1101101010 : 01100100000010001001111010101000	1101111111 : 0111110100101001100110010000001	1111100001 : 11000000100111101010000011001010
1101101011 : 1001101111101011011101100110011	1101111111 : 101101101001010011111001100111	1111100010 : 0101001000001100011000001010011
1101101100 : 0000001010010001111100000110001	1101111111 : 10111010100100000100111010000	1111100011 : 101001101111001110011110101100
1101101101 : 11111101100100010000011100110001	1101111111 : 10101010000110000000000110001	1111100011 : 101001101111001110011110101100
1101101110 : 01010111001101110101011001101	1101111111 : 1101011110011110110101010001110	1111100011 : 0000110001010010110100100000110
1101101111 : 10101000110001000101001100100	1101111111 : 010000101010110111001100010111	1111100101 : 111100111010110010010101111001
1101101111 : 10101000110001000101001100100	1101111111 : 1011101010010010000100111010000	1111100110 : 0101100111110001100111101010011
1101101111 : 10101000110001000101001100100	1101111111 : 10111010100100000100111010000	1111100111 : 101001101111001110011110101100
1101101111 : 10101000110001000101001100100	1101111111 : 10101010000110000100101010110001	1111100111 : 0011000001101110011110101100
1101101111 : 10101000110001000101001100100	1101111111 : 1101011110011110110101010001110	1111100111 : 0011000001101110011110101100

<98> 【표 1f】

1111101101	:	111110010100011100101011110110
1111101110	:	0101011000001001001111101011100
1111101111	:	1010100111110101100000010100011
1111110000	:	001111110011110101011011001010
1111110001	:	11000000011000001010100100110101
1111110010	:	0110101011001010000000111001111
1111110011	:	1001010100110101111110001100000
1111110100	:	00001100101011000110010111111001
1111110101	:	11110011010100111001101000000110
1111110110	:	01011001111110010011000010101100
1111110111	:	10100110000001101100111101010011
1111111000	:	00110000100100000101100111000101
1111111001	:	11001111011011111010011000111010
1111111010	:	01100101110001010000110010010000
1111111011	:	1001101000111010111100110110111
1111111100	:	000000111010001101101011110110
1111111101	:	11111100010111001001010100001001
1111111110	:	0101011011110110001111110100011
1111111111	:	10101001000010011100000001011100

<99> 도 9는 앞에서 개시한 부호기에 대응하는 복호기를 도시한다. 도 9를 살펴보면, 먼저 수신신호 $r(t)$ 가 입력되면 상기 수신신호 $r(t)$ 는 15개의 승산기 902, 904, ..., 930과 상관도 계산기 920에 입력된다. 그러면, 마스크 생성기 910은 15개의 모든 경우의 마스크를 생성하여 승산기 902, 904, ..., 906에 출력한다. 이 때, 승산기 902는 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 M_1 을 승산하여 상관도 계산기 922에 출력하고, 승산기 904는 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 M_2 를 승산하여 상관도 계산기 924에 출력한다. 이런 식으로 하여 승산기 906은 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 M_{15} 를 승산하여 상관도 계산기 926에 출력하면 수신신호 자신과 가능한 15개의 모든 마스크함수들이 수신신호 $r(t)$ 와 승산되어진, 총 16가지의 신호들이 16개의 상관도 계산기 920-926에 입력된다. 그러면, 상관도 계산기 920은 입력된 수신신호 $r(t)$ 를 32개의 모든 길이 32인 월시부호와 모든 월시부호의 심볼에 전부 -1(이진수의 경우 1)을 곱한 부호들에 따른 총 64가지와의 64가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때

계산되어진 월시부호 인덱스, 그리고, 상관도 계산기 인덱스인 0을 상관도 비교기 940에 출력한다. 이 때, 상기 상관도 계산기 인덱스는 상기 상관도 계산기에 입력된 신호가 수신신호에 몇 번째 마스크 함수가 승산되어져 입력되었는지를 나타내는 마스크 함수 인덱스와 동일하다. 그러나, 마스크 인덱스가 0이라 함은 아무런 마스크도 곱해지지 않았음을 의미한다. 그리고, 이와 동시에 상관도 계산기1 922는 입력된 수신신호 $r(t)$ 와 마스크 함수 M1을 승산한 신호를 32개의 모든 길이 32인 월시부호와 모든 월시부호의 심볼에 전부 -1(이진수의 경우 1)을 곱한 부호들에 따른 총 64가지와의 64가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 1을 상관도 비교기 940에 출력하고, 상관도 계산기2 924는 입력된 수신신호 $r(t)$ 와 마스크 함수 M2를 승산한 신호를 32개의 모든 길이 32인 월시부호와 모든 월시부호의 심볼에 전부 -1(이진수의 경우 1)을 곱한 부호들에 따른 총 64가지와의 64가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 2를 상관도 비교기 940에 출력하고, 이런 식으로 하여, 상관도 계산기15 926은 입력된 수신신호 $r(t)$ 와 마스크 함수 M15를 승산한 신호를 32개의 모든 길이 32인 월시부호와 모든 월시부호의 심볼에 전부 -1(이진수의 경우 1)을 곱한 부호들에 따른 총 64가지와의 64가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 15를 상관도 비교기 940에 출력한다.

<100> 그러면, 상관도비교기 940은 상기 상관도 계산기0-15 920 내지 926에서 입력된 16가지의 최대 상관값을 비교하여 가장 최대인 상관값을 구하면, 그 상관값에 대해 해당 상관도 계산기로부터 입력되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스,

즉, 수신신호 $r(t)$ 와 승산되어진 마스크함수의 인덱스를 가지고, 수신신호에 대한 복호신호를 계산하여 출력한다. 한편, 앞에서는 개시하고 있지 않으나 상기 상관도 계산기0-15 920 내지 926은 입력되는 데이터를 모든 월시부호와 비교함에 있어 직렬 또는 병렬 처리가 모두 가능하다.

<101> 도 10은 상기의 비교기 940의 동작을 플로우 차트로 나타낸 것이다. 도 10을 참조하여 설명하면 먼저 1000단계에서는 회수를 나타내는 인덱스 i 는 1로, 검색하려는 최대값, 월시부호 인덱스, 마스크인덱스들은 0으로 초기화한다. 그러면, 1010단계에서는 1번째 상관도 계산기 920에서 출력된 상관도의 1번째 최대값, 월시부호 인덱스, 마스크 인덱스들을 입력한다. 그러면, 1020에서 상기 1번째 최대값과 최대값을 비교하여 1번째 최대값이 크면 1030단계로 진행하여 최대값에 1번째 최대값을, 월시부호 인덱스에는 1번째 상관도 계산기로부터 입력된 월시부호 인덱스를, 마스크 인덱스에는 1번째 상관도 계산기로부터 입력된 마스크 인덱스를 차례로 저장하고, 1040단계를 진행하고, 1020에서 상기 1번째 최대값과 최대값을 비교하여 1번째 최대값이 작으면 1040단계를 바로 진행한다. 이 때, 1040단계에서는 회수를 나타내는 인덱스 i 가 상기 상관도계산기의 개수인 16인지를 판단한다. 1040단계에서는 회수를 나타내는 인덱스 i 가 상기 상관도계산기의 개수인 16이 아니면 1060단계를 진행하여 회수를 나타내는 인덱스 i 를 1씩 증가시키고, 다시 1010단계를 진행하여 2번째 상관도 계산기 922에서 출력된 상관도의 2번째 최대값, 월시부호 인덱스, 마스크 인덱스들을 입력하고 상기과 같은 과정을 반복한다. 상기과 같이 반복하다가 16번째 상관도 계산기로부터 입력된 상관값이 모두 비교되면 이때의 회수를 나타내는 i 가 16일 것이므로 1040단계에서 i 가 16인지를 판단하여 통과하면 1050단계로 진행하여 상기 변수인 월시부호 인덱스와 마스크 인덱스에 대응하는 복호비

트들을 출력한다.

<102> 상기의 제1실시 예는 전력제어의 단위인 슬롯의 개수가 16일 때 사용되는 (32,10) TFCI부호화기를 도시했다. 하기의 제2실시 예에서는 슬롯의 개수가 15일 때 사용되는 (30,10) TFCI부호화기로서 상기의 (32,10) TFCI부호화기를 통해 부호화된 32개의 부호화 심볼 중 2심볼을 천공하여 30심볼을 출력하는 부호화기를 도시한다.

<103> 제2실시 예

<104> 상기 제2실시 예에서는 상기의 생성방법에 따른 부호화 장치 및 방법을 제공한다. 상기의 부호화 장치의 구조는 제1실시 예의 부호화구조와 동일하다. 단, 1비트 발생기, 월시부호 발생기와 마스크 생성기에서 출력되는 수열들은 상기 제1실시 예에서의 수열들의 0번째와 16번째의 항을 천공함으로써 길이 30을 가진다. 따라서, 도 8을 참조하여 제2실시 예의 부호화기를 설명하면 10비트의 입력 정보비트들이 입력되면, 각각의 비트 $a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9$ 들은 승산기 840, 841, 842, 843, 844, 845, 846, 847, 848, 849에 입력된다. 이 때, 1비트 발생기 800에

서 항상 1인 심볼들이 승산기 840에 출력되고, 한 심볼이 승산기 840으로 입력될 때마다, 승산기 840에 입력된 a0과 승산되어져 가산기 860에 입력된다. 또, 월시부호 발생기 810에서는 길이 32인 월시부호 W1,W2,W4,W8,W16이 동시에 출력되는데, 제1월시부호 W1=1010101010101010101010101010101인 심볼들이 승산기 841에 출력되고, 한 심볼이 승산기 841로 입력될 때마다, 승산기 841에 입력된 a1과 승산되어져 가산기 860에 입력되고, 제2월시부호 W2=011001100110011011001100110011인 심볼들이 승산기 842에 출력되고, 한 심볼이 승산기 842로 입력될 때마다, 승산기 842에 입력된 a2와 승산되어져 가산기 860에 입력되고, 제4월시부호 W4=000111100001111000111100001111인 심볼들이 승산기 843에 출력되고, 한 심볼이 승산기 843으로 입력될 때마다, 승산기 843에 입력된 a3과 승산되어져 가산기 860에 입력되고, 제8월시부호 W8=000000011111111000000011111111인 심볼들이 승산기 844에 출력되고, 한 심볼이 승산기 844로 입력될 때마다, 승산기 844에 입력된 a4와 승산되어져 가산기 860에 입력되고, 제16월시부호 W16=000000000000000111111111111111인 심볼들이 승산기 845에 출력되고, 한 심볼이 승산기 845로 입력될 때마다, 승산기 845에 입력된 a5와 승산되어져 가산기 860에 입력된다.

<105> 마스크 생성기 820에서는 길이 30인 마스크 함수 M1,M2,M3,M4가 동시에 출력되는데, 제1마스크 함수 M1=010100001100011111000001110111인 심볼들이 승산기 846에 출력되고, 한 심볼이 승산기 846으로 입력될 때마다, 승산기 846에 입력된 a6과 승산되어져 가산기 860에 입력되고, 제2마스크 함수

M2=000000111001101110110111000111인 심볼들이 승산기 847에 출력되고, 한 심볼이 승산기 847로 입력될 때마다, 승산기 847에 입력된 a7과 승산되어져 가산기 860에 입력되고, 제4마스크 함수 M4=000 101011111001001101100101011인 심볼들이 승산기 848에 출력되고, 한 심볼이 승산기 848로 입력될 때마다, 승산기 848에 입력된 a8과 승산되어져 가산기 860에 입력되고, 제8마스크 함수 M8=0011100 00110111010111101010001인 심볼들이 승산기 849에 출력되고, 한 심볼이 승산기 849로 입력될 때마다, 승산기 849에 입력된 a9와 승산되어져 가산기 860에 입력된다.

<106> 그러면, 가산기 860은 승산기 840,841,842,843,844,845,846,847,848,849로부터 출력된 심볼들을 모두 가산하여 출력한다.

<107> 도 12은 상기 부호기의 동작에 대한 플로우 차트를 나타낸다. 도 12을 참조하면, 먼저 1100단계에서 10비트의 정보비트열 a_0, a_1, \dots, a_9 를 입력하고, 변수sum과 j을 0으로 초기화한다. 그러면, 1210단계를 진행하여 변수 j가 30인지를 판단하고, j가 30이 아니면 1220단계를 진행하여, 수열W1,W2,W4,W8,W16과 M1,M2,M3,M4의 j번째 심볼인, $W1(j), W2(j), W4(j), W8(j), W16(j)$ 과 $M1(j), M2(j), M3(j), M4(j)$ 를 입력한다. 그러면, 1230단계에서는 상기 1220단계에서 입력된 심볼들을 1200단계에서 입력되어진 정보비트와 심볼단위로 곱하여 전부 합하여 sum을 구한다. 그러면, 1240단계에서는 상기 1230단계에서 구해진 j번째 부호화 심볼인 sum을 출력하고, 1250으로 진행하여 j를 1만큼 증가시킨후 다시 1210단계를 진행한다. 그러면, 다시 1210에서는 심볼의 위치를 나타내는 j가 30인지를 판단하고 아니면 상기와 같은 동작을 반복하게 된다. 상기의 반복동작후 j가 30가 되면 상기의 부호화 동작을 완료하게 된다.

<108> 상기 (30,10)부호기는 상기 제1실시 예에 (32,10)부호기의 모든 부호어들에 대해서 1번째와 16번째 심볼들을 천공하여 만든 1024개의 부호어로 구성되어있다. 따라서, 총 부호어의 수는 1024개이다. 또한, 상기 1024개의 부호어들 중 32개의 길이 32인 월시부호에서 1번째와 16번째 심볼들이 천공된 길이 30인 월시부호와 상기 천공된 모든 월시부호의 심볼에 전부 1을 더한 실수의 경우 -1을 곱한, 총 64개의 부호와 상기 4개의 천공된 마스크함수 중 임의의 3개의 마스크의 조합에 구해지는 총8가지의 마스크함수의 조합으로 나타나는 (30,9)부호기, 상기 1024개의 부호어들 중 길이 30인 32개의 천공된 월시부호와 모든 천공된 월시부호의 심볼에 전부 1을 더한 실수의 경우 -1을 곱한, 총 64개의 부호와 상기 4개의 천공된 마스크함수 중 임의의 2개의 마스크의 조합에 구해지는 총8가지의 마스크함수의 조합으로 나타나는 (30,8)부호기, 상기 1024개의 부호어들 중 길이 30인 32개의 천공된 월시부호와 상기 모든 천공된 월시부호의 심볼에 전부 1을 더한 실수의 경우 -1을 곱한, 총 64개의 부호와 상기 4개의 상기 천공된 마스크함수중 임의의 1개의 마스크의 조합으로 나타나는 (30,7)부호기들은 모두 최소거리 10을 가진다.

<109> 상기와 같은 (30,9)부호기는 상기 도 7의 마스크함수 생성기에서 출력되는 4가지 마스크함수중 한 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있고, (30,8)부호기는 상기 도 7의 마스크함수 생성기에서 출력되는 4가지 마스크함수 중 두 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있다. 또한, (30,7)부호기는 상기 도 7의 마스크함수 생성기에서 출력되는 4가지 마스크함수중 세 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있다. 상기와 같은 작용을 함으로써 상기의 부호기는 입력정보 비트수에 따라 유동적으로 부호화가 가능하며, 부호기의 성능을 좌우하는 최소거리를 최대한 높임으로써 우수한 성능을 갖게 된다.

<110> 도 9는 상기 부호기에 따른 복호기를 도시한다. 도 9를 살펴보면, 먼저 수신신호 $r(t)$ 가 입력되면 상기 수신신호 $r(t)$ 는 15개의 승산기 902, 904, ..., 930과 상관도 계산기 920에 입력된다. 그러면, 마스크 생성기 910은 15개의 모든 경우의 마스크를 생성하여 승산기 902, 904, ..., 906에 출력한다. 이 때, 승산기 902는 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 $M1$ 을 승산하여 상관도 계산기 922에 출력하고, 승산기 904는 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 $M2$ 를 승산하여 상관도 계산기 924에 출력한다. 이런 식으로 하여 승산기 906은 입력된 수신신호 $r(t)$ 와 마스크 생성기 910으로부터 출력된 마스크함수 $M15$ 를 승산하여 상관도 계산기 926에 출력하면 수신신호 자신과 가능한 15개의 모든 마스크함수들이 수신신호 $r(t)$ 와 승산되어진, 총 16가지의 신호들이 16개의 상관도 계산기 920 내지 926에 입력된다. 그러면, 상관도 계산기 920은 입력된 수신신호 $r(t)$ 를 32개의 모든 길이 30인 월시부호와 모든 월시부호의 심볼에 전부 -1(이진수의 경우 1)을 곱한 부호들, 총 64가지와의 64가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 0을 상관도 비교기 940에 출력한다. 이 때, 상기 상관도 계산기의 인덱스는 상기 상관도 계산기에 입력된 신호가 수신신호에 몇 번째 마스크 함수가 승산되어져 입력되었는지를 나타내는 마스크 함수의 인덱스와 동일하다. 그러나, 마스크 인덱스가 0이라 함은 아무런 마스크도 곱해지지 않았음을 의미한다. 그리고, 이와 동시에 상관도 계산기 1은 입력된 수신신호 $r(t)$ 와 마스크 함수 $M1$ 을 승산한 신호를 32개의 모든 길이 30인 월시부호와 모든 월시부호의 심볼에 전부 -1(이진수의 경우 1)을 곱한 부호들, 총 64가지와의 64가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기

의 인덱스인 1을 상관도 비교기 940에 출력하고, 상관도 계산기2 924는 입력된 수신신호 $r(t)$ 와 마스크 함수 M2를 승산한 신호를 32개의 모든 길이 30인 월시부호와 모든 월시부호의 심볼에 전부 -1(이진수의 경우 1)을 곱한 부호들, 총 64가지와의 64가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 2를 상관도 비교기 940에 출력하고, 이런 식으로 하여, 상관도 계산기15 906은 입력된 수신신호 $r(t)$ 와 마스크 함수 M15를 승산한 신호를 32개의 모든 길이 30인 월시부호와 모든 월시부호의 심볼에 전부 -1(이진수의 경우 1)을 곱한 부호들, 총 64가지와의 64가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 15를 상관도 비교기 940에 출력한다.

<111> 그러면, 상관도 비교기 940은 상기 상관도 계산기 0-15 902 내지 906에서 입력된 16가지의 최대 상관값을 비교하여 가장 최대인 상관값을 구하면, 그 상관값에 대해 해당 상관도 계산기로부터 입력되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스, 즉, 수신신호 $r(t)$ 와 승산되어진 마스크함수의 인덱스를 가지고, 수신신호에 대한 복호신호를 계산하여 출력한다.

<112> 도 10은 상기의 비교기 940의 동작을 플로우 차트로 나타낸 것이다. 도 10을 참조하여 설명하면 먼저 1000단계에서는 회수를 나타내는 인덱스 I는 1로, 검색하려는 최대값, 월시부호 인덱스, 마스크인덱스들은 0으로 초기화한다. 그러면, 1010단계에서는 1번째 상관도 계산기에서 입력된 920에서 출력된 상관도의 1번째 최대값, 월시부호 인덱스, 마스크 인덱스들을 입력한다. 그러면, 1020에서 상기 1번째 최대값과 최대값을 비교하여 1번째 최대값이 크면 1030단계로 진행하여 최대값에 1번째 최대값을, 월시부호 인덱스

에는 1번째 상관도 계산기로부터 입력된 월시부호 인덱스를, 마스크 인덱스에는 1번째 상관도 계산기로부터 입력된 마스크 인덱스를 차례로 저장하고, 1040단계를 진행하고, 1020에서 상기 1번째 최대값과 최대값을 비교하여 1번째 최대값이 작으면 1040단계를 바로 진행한다. 이 때, 1040단계에서는 회수를 나타내는 인덱스 i 가 상기 상관도계산기의 개수인 16인지를 판단한다. 1040단계에서는 회수를 나타내는 인덱스 i 가 상기 상관도계산기의 개수인 16이 아니면 1060단계를 진행하여 회수를 나타내는 인덱스 i 를 1씩 증가시키고, 다시 1010단계를 진행하여 2번째 상관도 계산기에서 입력된 922에서 출력된 상관도의 2번째 최대값, 월시부호 인덱스, 마스크 인덱스들을 입력하고 상기와 같은 과정을 반복한다. 상기와 같이 반복하다가 16번째 상관도 계산기로부터 입력된 상관값이 모두 비교되면 이 때의 회수를 나타내는 i 가 16일 것이므로 1040단계에서 i 가 16인지를 판단하여 통과하면 1050단계로 진행하여 상기 변수인 월시부호 인덱스와 마스크 인덱스에 대응하는 복호비트들을 출력한다.

<113> 제3실시예

<114> 상기 제2실시예의 부호기는 상기 제1실시예의 부호어의 부호어를 1번째 심볼과 16번째 심볼을 천공하여 사용하는 부호기를 나타내었다. 그리고, 상기부호기는 마스크 생성기에서 4개의 마스크생성기의 입출력을 사용함으로써 (30,10)부호기로 사용할 수 있었고, (30,9)부호기는 상기 도 8의 마스크함수 생성기에서 출력되는 4가지 마스크함수 중 한 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있고, (30,8)부호기는 상기 도 8의 마스크함수 생성기에서 출력되는 4가지 마스크함수 중 두 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있고, 또한, (30,7)부호기는 상기 도 8의 마스크함수 생성기에서 출력되는 4가지 마스크함수 중 세 부분의 입력과 출력을 중단시킴으로써 구현되어

질 수 있다. 이 때, 상기와 같은 부호기에 있어서, (30,7-10)부호기의 최소거리는 10이다.

<115> 이하 설명하고자 하는 제3실시 예에서는 상기 제2실시 예에서 (30,7-10)부호기로 사용하는 경우 도 8의 1발생기의 입출력을 중단시키고, 이 대신 다른 하나의 마스크함수를 발생시킴으로써 (30,7-10)부호기의 최소거리가 11이 되도록 하는 부호기를 제공한다.

<116> 도 14는 상기와 같은 (30,7-10)부호기로 사용하는 경우 부호기의 최소거리가 11이 되도록 하는 부호기를 도시한다. 도 14에서 사용되는 마스크함수 M1, M2, M4, M8, M16은 다음과 같다.

<117> M1 = 000001011111000010110100111110

<118> M2 = 000110001100110001111010110111

<119> M4 = 010111100111101010000001100111

<120> M8 = 011011001000001111011100001111

<121> M16 = 100100011110011111000101010011

<122> 도 14를 참조하여 설명하면, (30,6)부호기로 사용하는 경우 스위치 1470이 상기 1 발생기 1400과 승산기 1440을 연결하고, 마스크생성기 1480의 모든 마스크함수들의 입출력을 중단하면서, 그 이후의 동작은 상기 제2실시 예와 유사하다. 상기 부호기가 (30,7-10)부호기로 사용이되면, 스위치 1470은 상기 마스크생성기 1480과 승산기 1440을 연결하고, 7-10 비트의 입력 정보비트들이 입력되면, 각각의 비트 a0, a1, a2, a3, a4, a5, a6, a7, a8, a9들은 승산기 1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449에 입력된다. 이 때, 마스크생성기 1480에서 마스크함수 M16 =

100100011110011111000101010011인 심볼들이 승산기 1440에 출력되고, 한 심볼이 승산기 1440으로 입력될 때마다, 승산기 1440에 입력된 a0과 승산되어져 가산기 1460에 입력된다. 또, 월시부호 발생기 1410에서는 길이 30인 월시부호 W1,W2,W4,W8,W16이 동시에 출력되는데, 제1월시부호 W1=101010101010101101010101010101인 심볼들이 승산기 1441에 출력되고, 한 심볼이 승산기 1441로 입력될 때마다, 승산기 1441에 입력된 a1과 승산되어져 가산기 1460에 입력된다. 제2월시부호 W2=011001100110011011001100110011인 심볼들이 승산기 1442에 출력되고, 한 심볼이 승산기 1442로 입력될 때마다, 승산기 1442에 입력된 a2와 승산되어져 가산기 1460에 입력된다. 제4월시부호 W4=000111100001111000111100001111인 심볼들이 승산기 1443에 출력되고, 한 심볼이 승산기 1443으로 입력될 때마다, 승산기 1443에 입력된 a3과 승산되어져 가산기 1460에 입력된다. 제8월시부호 W8=000000011111111000000011111111인 심볼들이 승산기 1444에 출력되고, 한 심볼이 승산기 1444로 입력될 때마다, 승산기 1444에 입력된 a4와 승산되어져 가산기 1460에 입력된다. 제16월시부호 W16=000000000000000011111111111111인 심볼들이 승산기 1445에 출력되고, 한 심볼이 승산기 1445로 입력될 때마다, 승산기 1445에 입력된 a5와 승산되어져 가산기 1460에 입력된다.

<123> 마스크 생성기 1420에서는 길이 30인 마스크 함수 M1,M2,M3,M4가 동시에 출력되는데, 제1마스크 함수 M1=000001011111000010110100111110인 심볼들이 승산기 1446에 출력되고, 한 심볼이 승산기 1446으로 입력될 때마다, 승산기 1446에 입력된 a6과 승산되어져 가산기 1460에 입력된다. 제2마스크 함수 M2=000110001100110001111010110111인 심볼들이 승산기 1447에 출력되고, 한 심볼이 승산기 1447로 입력될 때마다, 승산기 1447에 입력된 a7과 승산되어져 가산기 1460에 입력된다. 제4마스크 함수 M4 =

010111100111101010000001100111인 심볼들이 승산기 1448에 출력되고, 한 심볼이 승산기 1448로 입력될 때마다, 승산기 1448에 입력된 a8과 승산되어져 가산기 1460에 입력된다. 제8마스크 함수 $M8 = 011011001000001111011100001111$ 인 심볼들이 승산기 1449에 출력되고, 한 심볼이 승산기 1449로 입력될 때마다, 승산기 1449에 입력된 a9와 승산되어져 가산기 1460에 입력된다.

<124> 그러면, 가산기 1460은 승산기

1440, 1441, 1442, 1443, 1444, 1445, 1446, 1447, 1448, 1449로부터 출력된 심볼들을 모두 가산하여 출력한다.

<125> 상기 (30,10)부호기는 32개의 길이 32인 월시부호에서 1번째와 16번째 심볼들이 천공된 길이 30인 월시부호와 상기 천공된 모든 월시부호의 심볼에 전부 1을 더한 실수의 경우 -1을 곱한, 총 64개의 부호와 상기 4개의 마스크함수의 조합에 구해지는 총 16가지의 마스크함수의 조합으로 나타나 총 부호어의 수는 1024개이다. 또한, 32개의 길이 30인 월시부호에서 1번째와 16번째 심볼들이 천공된 길이 30인 월시부호와 상기 5개의 천공된 마스크함수중 임의의 4개의 마스크의 조합에 구해지는 총 16가지의 마스크함수의 조합으로 나타나는 (30,9)부호기, 길이 30인 32개의 천공된 월시부호와 상기 5개의 마스크함수중 임의의 3개의 마스크의 조합에 구해지는 총 8가지의 마스크함수의 조합으로 나타나는 (30,8)부호기, 상기 1024개의 부호어들 중 길이 30인 32개의 천공된 월시부호와 상기 5개의 상기 천공된 마스크함수중 임의의 2개의 마스크의 조합으로 나타나는 (30,7)부호기들은 모두 최소거리 11을 가진다.

<126> 상기와 같은 (30,9)부호기는 상기 도 14의 마스크함수 생성기 1420에서 출력되는 4가지 마스크함수중 한 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있고,

(30,8)부호기는 상기 도 14의 마스크함수 생성기 1420에서 출력되는 4가지 마스크함수 중 두 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있고, 또한, (30,7)부호기는 상기 도 14의 마스크함수 생성기 1420에서 출력되는 4가지 마스크함수 중 세 부분의 입력과 출력을 중단시킴으로써 구현되어질 수 있다.

<127> 도 16은 상기 제3실시예에 따른 부호기의 동작에 대한 플로우 차트를 나타낸다. 도 16을 참조하면, 먼저 1600단계에서 10비트의 정보비트열 a_0, a_1, \dots, a_9 를 입력하고, 변수sum과 j을 0으로 초기화한다. 그러면, 1610단계를 진행하여 변수 j가 30인지를 판단하고, j가 30이 아니면 1620단계를 진행하여, 수열 $W_1, W_2, W_4, W_8, W_{16}$ 과 M_1, M_2, M_3, M_4, M_5 의 j번째 심볼인, $W_1(j), W_2(j), W_4(j), W_8(j), W_{16}(j)$ 과 $M_1(j), M_2(j), M_3(j), M_4(j), M_5(j)$ 를 입력한다. 그러면, 1630단계에서는 상기 1620단계에서 입력된 심볼들을 1600단계에서 입력되어진 정보비트와 심볼단위로 곱하여 전부 합하여 sum을 구한다. 그러면, 1640단계에서는 상기 1630단계에서 구해진 j번째 부호화 심볼인 sum을 출력하고, 1650으로 진행하여 j를 1만큼 증가시킨후 다시 1610단계를 진행한다. 그러면, 다시 1610에서는 심볼의 위치를 나타내는 j가 30인지를 판단하고 아니면 상기와 같은 동작을 반복하게 된다. 상기의 반복동작후 j가 30가 되면 상기의 부호화 동작을 완료하게 된다.

<128> 도 15는 상기 부호기에 따른 복호기를 도시한다. 도 15를 살펴보면, 먼저 수신신호 $r(t)$ 가 입력되면 상기 수신신호 $r(t)$ 는 31개의 승산기 1502, 1504, ..., 1506과 상관도 계산기 1520에 입력된다. 그러면, 마스크 생성기 1510은 31개의 모든 경우의 마스크를 생성하여 승산기 1502, 1504, ..., 1506에 출력한다. 이 때, 승산기 1502는 입력된 수신신호 $r(t)$ 와 마스크 생성기 1510으로부터 출력된 마스크함수 M_1 을 승산하여 상관도 계산기 1522에 출력하고, 승산기 1504는 입력된 수신신호 $r(t)$ 와 마스크 생성기 1510으로부터

출력된 마스크함수 M2를 승산하여 상관도 계산기2 1524에 출력한다. 이런 식으로 하여 승산기 1506은 입력된 수신신호 $r(t)$ 와 마스크 생성기 1510으로부터 출력된 마스크함수 M31를 승산하여 상관도 계산기 1526에 출력하면 수신신호 자신과 가능한 31개의 모든 마스크함수들이 수신신호 $r(t)$ 와 승산되어진, 총 32가지의 신호들이 32개의 상관도 계산기 0-32 1520 내지 1526에 입력된다. 그러면, 상관도 계산기0 1520은 입력된 수신신호 $r(t)$ 를 32개의 모든 길이 30인 월시부호들과의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 0을 상관도 비교기31 1540에 출력한다. 이 때, 상기 상관도 계산기의 인덱스는 상기 상관도 계산기에 입력된 신호가 수신신호에 몇 번째 마스크 함수가 승산되어져 입력되었는지를 나타내는 마스크 함수의 인덱스와 동일하다. 그러나, 마스크 인덱스가 0이라 함은 아무런 마스크도 곱해지지 않았음을 의미한다. 그리고, 이와 동시에 상관도 계산기1은 입력된 수신신호 $r(t)$ 와 마스크 함수 M1을 승산한 신호를 32개의 모든 길이 30인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 1을 상관도 비교기 1540에 출력하고, 상관도 계산기2 1524는 입력된 수신신호 $r(t)$ 와 마스크 함수 M2를 승산한 신호를 32개의 모든 길이 30인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 2를 상관도 비교기 1540에 출력하고, 이런 식으로 하여, 상관도 계산기31 1506은 입력된 수신신호 $r(t)$ 와 마스크 함수 M31를 승산한 신호를 32개의 모든 길이 30인 월시부호와의 32가지 상관도를 구하여 상관도의 값이 가장 큰 값과, 그 때 계산되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스인 31을 상관도 비교기 1540에 출력한다.

<129> 그러면, 상관도 비교기 1540은 상기 상관도 계산기 0-31 1502 내지 1506에서 입력된 32가지의 최대 상관값을 비교하여 가장 최대인 상관값을 구하면, 그 상관값에 대해 해당 상관도 계산기로부터 입력되어진 월시부호의 인덱스, 그리고, 상관도 계산기의 인덱스, 즉, 수신신호 $r(t)$ 와 송산되어진 마스크함수의 인덱스를 가지고, 수신신호에 대한 복호신호를 계산하여 출력한다.

【발명의 효과】

<130> 상기와 같이 본 발명은 하나의 하드웨어를 통해 기본적인 TFCI의 오류정정부호화 방식과 서비스의 확장을 위하여 TFCI가 확장됨에 따른 오류정정부호화 방식을 모두 구현할 수 있어 입력정보 비트 수에 따라 유동적으로 부호화가 가능한 장점이 있다. 또한, 부호기의 성능을 좌우하는 최소거리를 최대한 높임으로써 우수한 성능을 갖는 부호분할 다중접속시스템을 구현할 수 있는 효과가 있다.

【특허청구범위】**【청구항 1】**

부호분할다중접속시스템에 있어서,

기본 데이터를 월시부호와 승산하여 출력하는 월시부호화기와,

확장 데이터의 비트 수에 의해 마스크함수의 사용 개수를 결정하고, 상기 결정된 사용 개수의 마스크함수와 상기 확장 데이터를 승산하여 출력하는 마스크함수 부호화기와,

상기 월시부호화기와 상기 마스크함수 부호화기의 출력을 가산하는 가산기로 구성함을 특징으로 하는 부호화장치.

【청구항 2】

제1항에 있어서, 상기 월시부호화기에서 사용되는 월시부호는, 월시번호 1, 월시번호 2, 월시번호 4, 월시번호 8 및 월시번호 16에 해당하는 월시부호임을 특징으로 하는 부호화장치.

【청구항 3】

제1항에 있어서, 상기 마스크함수 부호화기는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 구하여진 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크함수들 중 기저 마스크함수

만을 선택적으로 생성함을 특징으로 하는 부호화장치.

【청구항 4】

제3항에 있어서,

상기 확장 데이터의 비트 수에 의해 최대 결정될 수 있는 기저 마스크함수는

00101000011000111111000001110111의 제1마스크 함수와,

00000001110011010110110111000111의 제2마스크 함수와,

00001010111110010001101100101011의 제3마스크 함수와,

00011100001101110010111101010001의 제4마스크 함수로 구성됨을 특징으로 하는 부호화장치.

【청구항 5】

제4항에 있어서, 상기 마스크함수 부호화기는,

상기 확장 데이터의 비트 수에 비례하여 상기 제1마스크함수 내지 제4마스크함수의 사용 여부를 결정함을 특징으로 하는 부호화장치.

【청구항 6】

부호분할다중접속시스템에 있어서,

심볼 값이 1인 제1심볼을 연속적으로 생성하는 제1심볼발생기와,

소정 길이를 가지는 제2심볼들을 생성하는 제2심볼발생기와,

소정의 m 시퀀스에 의해 상기 소정 길이를 가지는 기저 마스크함수들을 생성하는
마스크생성기와,

입력 데이터를 상기 제1, 제2심볼들 및 상기 기저 마스크함수들과 승산하는 승산
기들과,

상기 승산기들의 출력을 상기 소정 길이 단위로 가산하여 부호심볼을 출력하는 가
산기로 구성함을 특징으로 하는 부호화장치.

【청구항 7】

제6항에 있어서, 상기 승산기들은,

상기 소정 길이의 제1심볼을 상기 입력 데이터의 제1비트에 의해 승산하는 제1승
산기와,

상기 소정 길이의 제2심볼들을 상기 입력 데이터의 제2비트 내지 제6비트에 의해
승산하는 제2 내지 제6승산기와,

상기 기저 마스크함수들을 상기 입력 데이터를 구성하는 확장 데이터의 비트 수에
대응하는 기저 마스크함수만을 승산하는 제7 내지 제10승산기로 구성함을 특징으로 하는
부호화장치.

【청구항 8】

제6항 또는 제7항에 있어서, 상기 제2심볼들이, 월시번호 1, 월시번호 2, 월시번호 4, 월시번호 8 및 월시번호 16을 가지는 월시부호임을 특징으로 하는 부호화장치.

【청구항 8】

【청구항 9】

제6항 또는 제7항에 있어서, 상기 마스크생성기는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크함수들 중 기저 마스크함수만을 선택적으로 생성함을 특징으로 하는 부호화장치.

【청구항 10】

제6항에 있어서, 상기 입력 데이터가 전송율정보임을 특징으로 하는 부호화장치.

【청구항 11】

제10항에 있어서, 상기 소정 길이가 32임을 특징으로 하는 부호화장치.

【청구항 12】

제9항에 있어서, 상기 기저 마스크함수는, 00101000011000111111000001110111의 제1마스크 함수와, 00000001110011010110110111000111의 제2마스크 함수와, 00001010111110010001101100101011의 제3마스크 함수와,

00011100001101110010111101010001의 제4마스크 함수로 구성됨을 특징으로 하는 부호화 장치.

【청구항 13】

부호분할다중접속시스템에 있어서,

소정의 m 시퀀스에 의해 생성되는 소정 길이의 기저 마스크 함수들을 조합하여 마스크함수를 생성하는 마스크생성기와,

상기 마스크생성기로부터 제공되는 마스크함수와 수신신호를 승산하는 승산기들과,

상기 수신신호와 소정 길이의 월시부호들간의 상관도를 계산하여 가장 큰 상관도 값과, 상기 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스를 제공하는 상관도계산기와,

상기 승산기들의 출력과 상기 소정 길이의 월시부호들간의 상관도를 계산하여 가장 큰 상관도 값과, 상기 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스를 제공하는 상관도계산기들과,

상기 상관도계산기들로부터 제공되는 상관도 값들을 비교하여 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스에 대응하는 입력정보비트를 출력하는 상관도 비교기로 구성함을 특징으로 하는 복호화장치.

【청구항 14】

제13항에 있어서, 상기 마스크생성기는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 구하여진 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크 함수들 중 기저 마스크 함수만을 선택하여 상기 선택된 기저 마스크 함수의 조합에 의해 15개의 마스크함수를 생성함을 특징으로 하는 복호화장치.

【청구항 15】

제14항에 있어서, 상기 월시부호들은,
길이 32를 가지는 32개의 월시부호와, 상기 32개의 월시부호에 -1을 곱함으로서 얻어지는 32개의 월시부호를 합한 64개의 월시부호로 구성됨을 특징으로 하는 복호화장치.

【청구항 16】

부호분할다중접속시스템에 있어서,
심볼 값이 1인 제1심볼을 연속적으로 생성하는 과정과,
소정 길이를 가지는 제2심볼들을 생성하는 과정과,
소정의 m 시퀀스에 의해 상기 소정 길이를 가지는 기저 마스크함수들을 생성하는 과정과,
입력 데이터를 상기 제1, 제2심볼들 및 상기 기저 마스크함수들과 승산하는 과정과

상기 승산기들의 출력을 상기 소정 길이 단위로 가산하여 부호심볼을 출력하는 과정으로 이루어짐을 특징으로 하는 부호화방법.

【청구항 17】

제16항에 있어서, 상기 승산하는 과정은,

상기 소정 길이의 제1심볼과 상기 소정 길이의 제2심볼들을 상기 입력 데이터를 구성하는 기본 데이터의 비트들에 의해 승산하는 과정과,

상기 기저 마스크함수들을 상기 입력 데이터를 구성하는 확장 데이터의 비트 수에 대응하는 기저 마스크함수만을 승산하는 과정으로 이루어짐을 특징으로 하는 부호화장치

【청구항 18】

제16항 또는 제17항에 있어서, 상기 제2심볼들이, 월시번호 1, 월시번호 2, 월시번호 4, 월시번호 8 및 월시번호 16을 가지는 월시부호임을 특징으로 하는 부호화방법.

【청구항 19】

제16항 또는 제17항에 있어서, 상기 기저 마스크함수는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크함수들 중 선택적으로 생성된 기저 마

스크함수임을 특징으로 하는 부호화방법.

【청구항 20】

제16항에 있어서, 상기 입력 데이터가 전송율정보임을 특징으로 하는 부호화방법.

【청구항 21】

제20항에 있어서, 상기 소정 길이가 32임을 특징으로 하는 부호화방법.

【청구항 22】

제19항에 있어서, 상기 기저 마스크함수는, 00101000011000111111000001110111의 제1마스크 함수와, 00000001110011010110110111000111의 제2마스크 함수와, 00001010111110010001101100101011의 제3마스크 함수와, 00011100001101110010111101010001의 제4마스크 함수로 이루어짐을 특징으로 하는 부호화방법.

【청구항 23】

부호분할다중접속시스템에 있어서,

소정의 m 시퀀스에 의해 생성되는 소정 길이의 기저 마스크 함수들을 조합하여 마스크함수를 생성하는 과정과,

상기 마스크함수와 수신신호를 승산하는 과정과,

상기 수신신호와 소정 길이의 월시부호들간의 상관도를 계산하여 가장 큰 상관도 값과, 상기 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스를 제공하는 과정과,

상기 송신하는 과정에 따른 출력과 상기 소정 길이의 월시부호들간의 상관도를 계산하여 가장 큰 상관도 값과, 상기 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스를 제공하는 과정과,

상기 제공되는 상관도 값들을 비교하여 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스에 대응하는 입력정보비트를 출력하는 과정으로 이루어짐을 특징으로 하는 복호화방법.

【청구항 24】

제23항에 있어서, 상기 마스크함수는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 구하여진 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크 함수들 중 기저 마스크 함수만을 선택하여 상기 선택된 기저 마스크 함수의 조합에 의해 생성되는 15개의 마스크 함수임을 특징으로 하는 복호화방법.

【청구항 25】

제24항에 있어서, 상기 월시부호들은,

길이 32를 가지는 32개의 월시부호와, 상기 32개의 월시부호에 -1을 곱함으로써 얻어지는 32개의 월시부호를 합한 64개의 월시부호로 이루어짐을 특징으로 하는 복호화방법.

【청구항 26】

부호분할다중접속시스템에 있어서,
 심볼 값이 1인 제1심볼을 연속적으로 생성하는 제1심볼발생기와,
 제2심볼들의 불특정한 두 심볼을 천공하여 소정 길이를 가지는 천공된 제2심볼을 생성하는 제2심볼발생기와,
 소정의 m 시퀀스에 의해 생성되는 기저 마스크함수들의 불특정한 두 심볼을 천공하여 상기 소정 길이를 가지는 천공 기저 마스크함수를 생성하는 마스크생성기와,
 입력 데이터를 상기 제1심볼, 천공된 제2심볼들 및 상기 천공 기저 마스크함수들과 승산하는 승산기들과,
 상기 승산기들의 출력을 상기 소정 길이 단위로 가산하여 부호심볼을 출력하는 가산기로 구성함을 특징으로 하는 부호화장치.

【청구항 27】

제26항에 있어서, 상기 승산기들은,
 상기 소정 길이의 제1심볼을 상기 입력 데이터의 제1비트에 의해 승산하는 제1승산기와,

상기 소정 길이의 제2심볼들을 상기 입력 데이터의 제2비트 내지 제6비트에 의해
 승산하는 제2 내지 제6승산기와,

상기 기저 마스크함수들을 상기 입력 데이터를 구성하는 확장 데이터의 비트 수에
 대응하는 기저 마스크함수만을 승산하는 제7 내지 제10승산기로 구성함을 특징으로 하는
 부호화장치.

【청구항 28】

제26항 또는 제27항에 있어서, 상기 제2심볼들이, 월시번호 1, 월시번호 2, 월시번호 4, 월시번호 8 및 월시번호 16을 가지는 월시부호임을 특징으로 하는 부호화장치.

【청구항 29】

제26항 또는 제27항에 있어서, 상기 마스크생성기는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크함수들 중 기저 마스크함수만을 선택한 후 불특정한 두 심볼을 천공함으로서 천공 기저 마스크함수를 생성함을 특징으로 하는 부호화장치.

【청구항 30】

제26항에 있어서, 상기 입력 데이터가 전송율정보임을 특징으로 하는 부호화장치.

【청구항 31】

제30항에 있어서, 상기 소정 길이가 30임을 특징으로 하는 부호화장치.

【청구항 32】

제29항에 있어서, 상기 천공 기저 마스크함수는, 010100001100011111000001110111의 제1마스크 함수와, 000000111001101110110111000111의 제2마스크 함수와, 000101011111001001101100101011의 제3마스크 함수와, 001110000110111010111101010001의 제4마스크 함수로 구성됨을 특징으로 하는 부호화장치.

【청구항 33】

부호분할다중접속시스템에 있어서,

소정의 m 시퀀스에 의해 생성되는 기저 마스크 함수들의 조합에 의해 결정되는 마스크함수들의 불특정한 두 심볼을 천공하여 소정 길이를 가지는 천공 마스크함수를 생성하는 마스크생성기와,

상기 소정 길이의 마스크함수들과 수신신호를 승산하는 승산기들과,

상기 수신신호와 불특정한 두 심볼이 천공된 소정 길이의 월시부호들간의 상관도를 계산하여 가장 큰 상관도 값과, 상기 가장 큰 상관도 값에 대응하는 월시부호 인덱스를

및 마스크함수 인덱스를 제공하는 상관도계산기와,

상기 승산기들의 출력과 상기 소정 길이의 월시부호들간의 상관도를 계산하여 가장 큰 상관도 값과, 상기 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스를 제공하는 상관도계산기들과,

상기 상관도계산기들로부터 제공되는 상관도 값들을 비교하여 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스에 대응하는 입력정보비트를 출력하는 상관도 비교기로 구성함을 특징으로 하는 복호화장치.

【청구항 34】

제33항에 있어서, 상기 마스크생성기는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 구하여진 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크 함수들 중 기저 마스크 함수만을 선택하여 상기 선택된 기저 마스크 함수의 조합에 의해 15개의 마스크함수를 생성한 후 불특정한 두 심볼을 천공하여 천공 마스크함수를 생성함을 특징으로 하는 복호화장치.

【청구항 35】

제34항에 있어서, 상기 월시부호들은,

길이 30을 가지는 32개의 월시부호와, 상기 32개의 월시부호에 -1을 곱함으로서 얻어지는 32개의 월시부호를 합한 64개의 월시부호로 구성됨을 특징으로 하는 복호화장치.

【청구항 36】

부호분할다중접속시스템에 있어서,
 심볼 값이 1인 제1심볼을 연속적으로 생성하는 과정과,
 제2심볼들의 불특정한 두 심볼을 천공하여 소정 길이를 가지는 천공된 제2심볼을 생성하는 과정과,
 소정의 m 시퀀스에 의해 생성되는 기저 마스크함수들의 불특정한 두 심볼을 천공하여 상기 소정 길이를 가지는 천공 기저 마스크함수를 생성하는 과정과,
 입력 데이터를 상기 제1심볼, 천공된 제2심볼들 및 상기 천공 기저 마스크함수들과 승산하는 과정과,
 상기 승산기들의 출력을 상기 소정 길이 단위로 가산하여 부호심볼을 출력하는 과정으로 이루어짐을 특징으로 하는 부호화방법.

【청구항 37】

제36항에 있어서, 상기 승산하는 과정은,
 상기 소정 길이의 제1심볼을 상기 입력 데이터의 제1비트에 의해 승산하는 과정과,

상기 소정 길이의 제2심볼들을 상기 입력 데이터의 제2비트 내지 제6비트에 의해
승산하는 과정과,

상기 기저 마스크함수들을 상기 입력 데이터를 구성하는 확장 데이터의 비트 수에
대응하는 기저 마스크함수만을 승산하는 과정으로 이루어짐을 특징으로 하는 부호화방법

【청구항 38】

제36항 또는 제37항에 있어서, 상기 제2심볼들이, 월시번호 1, 월시번호 2, 월시번호 4, 월시번호 8 및 월시번호 16을 가지는 월시부호임을 특징으로 하는 부호화방법.

【청구항 39】

제36항 또는 제37항에 있어서, 상기 천공 기저 마스크함수는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크함수들 중 기저 마스크함수만을 선택한 후 불특정한 두 심볼을 천공함으로서 생성되는 천공 기저 마스크함수임을 특징으로 하는 부호화방법.

【청구항 40】

제36항에 있어서, 상기 입력 데이터가 전송율정보임을 특징으로 하는 부호화방법.

【청구항 41】

제40항에 있어서, 상기 소정 길이가 30임을 특징으로 하는 부호화방법.

【청구항 42】

제39항에 있어서, 상기 천공 기저 마스크함수는, 010100001100011111000001110111의 제1마스크 함수와, 000000111001101110110111000111의 제2마스크 함수와, 000101011111001001101100101011의 제3마스크 함수와, 001110000110111010111101010001의 제4마스크 함수로 이루어짐을 특징으로 하는 부호화방법.

【청구항 43】

부호분할다중접속시스템에 있어서,

소정의 m 시퀀스에 의해 생성되는 기저 마스크 함수들의 조합에 의해 결정되는 마스크함수들의 불특정한 두 심볼을 천공하여 소정 길이를 가지는 천공 마스크함수를 생성하는 과정과,

상기 소정 길이의 마스크함수들과 수신신호를 승산하는 과정과,

상기 수신신호와 불특정한 두 심볼이 천공된 소정 길이의 월시부호들간의 상관도를 계산하여 가장 큰 상관도 값과, 상기 가장 큰 상관도 값에 대응하는 월시부호 인덱스

및 마스크함수 인덱스를 제공하는 과정과,

상기 승산하는 과정에 따른 출력과 상기 소정 길이의 월시부호들간의 상관도를 계산하여 가장 큰 상관도 값과, 상기 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스를 제공하는 과정과,

상기 제공되는 상관도 값들을 비교하여 가장 큰 상관도 값에 대응하는 월시부호 인덱스 및 마스크함수 인덱스에 대응하는 입력정보비트를 출력하는 과정으로 이루어짐을 특징으로 하는 복호화방법.

【청구항 44】

제43항에 있어서, 상기 천공 마스크함수는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 구하여진 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크 함수들 중 기저 마스크 함수만을 선택하여 상기 선택된 기저 마스크 함수의 조합에 의해 15개의 마스크함수를 생성한 후 불특정한 두 심볼을 천공함으로서 생성되는 천공 마스크함수임을 특징으로 하는 복호화방법.

【청구항 45】

제44항에 있어서, 상기 월시부호들은,

길이 30을 가지는 32개의 월시부호와, 상기 32개의 월시부호에 -1을 곱함으로서 얻어지는 32개의 월시부호를 합한 64개의 월시부호로 구성됨을 특징으로 하는 복호화방법.

【청구항 46】

부호분할다중접속시스템에 있어서,

심볼 값이 1인 제1심볼을 연속적으로 생성하는 제1심볼발생기와,

제2심볼들의 첫 번째 심볼과 상기 첫 번째 심볼을 제외한 나머지 심볼들 중 불특정한 한 심볼을 천공하여 소정 길이를 가지는 천공된 제2심볼을 생성하는 제2심볼발생기와,

소정의 m 시퀀스에 의해 생성되는 기저 마스크함수들 중 어느 한 기저 마스크함수의 첫 번째 심볼과 상기 첫 번째 심볼을 제외한 나머지 심볼들 중 불특정한 한 심볼을 천공하여 소정 길이를 가지는 천공 기저 마스크함수를 생성하는 제1마스크생성기와,

상기 제1마스크생성기에 의해 사용되지 않는 상기 기저 마스크함수들의 첫 번째 심볼과 상기 첫 번째 심볼을 제외한 나머지 심볼들 중 불특정한 한 심볼을 천공하여 소정 길이를 가지는 천공 기저 마스크함수들을 생성하는 제2마스크생성기와,

입력 데이터로 기본 데이터만 입력되면 상기 제1심볼발생기를 선택하고, 상기 입력 데이터로 기본 데이터와 확장 데이터가 입력되면 상기 제1마스크생성기를 선택하는 스위치와,

상기 입력 데이터를 상기 스위치와 상기 제2심볼발생기 및 상기 제2마스크생성기로 부터 제공되는 심볼들과 승산하는 승산기들과,

상기 승산기들의 출력을 상기 소정 길이 단위로 가산하여 부호심볼을 출력하는 가산기로 구성함을 특징으로 하는 부호화장치.

【청구항 47】

제46항에 있어서, 상기 승산기들은,

상기 스위치를 통해 제공되는 상기 소정 길이의 심볼을 상기 입력 데이터의 제1비트에 의해 승산하는 제1승산기와,

상기 소정 길이의 천공된 제2심볼들을 상기 기본 데이터에 의해 승산하는 제2 내지 제6승산기와,

상기 제2마스크생성기로부터 생성되는 천공 기저 마스크함수들을 상기 확장 데이터의 비트 수에 대응하는 천공 기저 마스크함수만을 승산하는 제7 내지 제10승산기로 구성함을 특징으로 하는 부호화장치.

【청구항 48】

제46항 또는 제47항에 있어서, 상기 제2심볼들이, 월시번호 1, 월시번호 2, 월시번호 4, 월시번호 8 및 월시번호 16을 가지는 월시부호임을 특징으로 하는 부호화장치.

【청구항 49】

제46항 또는 제47항에 있어서, 상기 마스크생성기는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스

를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 열처리 함수의 역함수에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크함수들 중 기저 마스크함수만을 선택한 후 첫 번째 심볼과 불특정한 한 심볼을 천공함으로서 천공 기저 마스크함수를 생성함을 특징으로 하는 부호화장치.

【청구항 50】

제25항에 있어서, 상기 입력 데이터가 전송율정보임을 특징으로 하는 부호화장치.

【청구항 51】

제50항에 있어서, 상기 소정 길이가 30임을 특징으로 하는 부호화장치.

【청구항 52】

제49항에 있어서, 상기 천공 기저 마스크함수는, 010100001100011111000001110111의 제1마스크 함수와, 000000111001101110110111000111의 제2마스크 함수와, 000101011111001001101100101011의 제3마스크 함수와, 001110000110111010111101010001의 제4마스크 함수로 구성됨을 특징으로 하는 부호화장치.

【청구항 53】

제52항에 있어서, 상기 제1마스크생성기로부터 발생하는 기저 마스크 함수는 10010001111100111111000101010011임을 특징으로 하는 부호화장치.

【청구항 53】

【청구항 54】

부호분할다중접속시스템에 있어서,

심볼 값이 1인 제1심볼을 연속적으로 생성하는 과정과,

제2심볼들의 첫 번째 심볼과 상기 첫 번째 심볼을 제외한 나머지 심볼들 중 불특정한 한 심볼을 천공하여 소정 길이를 가지는 천공된 제2심볼을 생성하는 과정과,

소정의 m 시퀀스에 의해 생성되는 제1기저 마스크함수의 첫 번째 심볼과 불특정한 한 심볼을 천공하여 소정 길이를 가지는 제1천공 기저 마스크함수를 생성하는 과정과,

상기 제2 내지 제5기저 마스크함수들의 첫 번째 심볼과 불특정한 한 심볼을 천공하여 소정 길이를 가지는 제2 내지 제5천공 기저 마스크함수들을 생성하는 과정과,

입력 데이터로 기본 데이터만 입력되면 상기 제1심볼을 선택하고, 상기 입력 데이터로 기본 데이터와 확장 데이터가 입력되면 상기 제1천공 기저 마스크함수를 선택하는 과정과,

상기 입력 데이터를 상기 선택되는 제1심볼 또는 제1천공 기저 마스크함수와 상기 제2심볼 및 상기 제2 내지 제5천공 기저 마스크함수들을 승산하는 과정과,

상기 승산 출력을 상기 소정 길이 단위로 가산하여 부호심볼을 출력하는 과정으로 이루어짐을 특징으로 하는 부호화방법.

【청구항 55】

제54항에 있어서, 상기 승산하는 과정은,

상기 소정 길이를 가지는 상기 선택되는 제1심볼 또는 제1천공 기저 마스크함수를
상기 기본 데이터의 제1비트에 의해 승산하는 과정과,

상기 소정 길이의 천공된 제2심볼들을 상기 기본 데이터의 제2비트 내지 제7비트에
의해 승산하는 과정과,

상기 제2 내지 제5천공 기저 마스크함수들 중 상기 확장 데이터의 비트 수에 대응
하는 천공 기저 마스크함수만을 승산하는 과정으로 이루어짐을 특징으로 하는 부호화방
법.

【청구항 56】

제54항 또는 제55항에 있어서, 상기 제2심볼들이, 월시번호 1, 월시번호 2, 월시번
호 4, 월시번호 8 및 월시번호 16을 가지는 월시부호임을 특징으로 하는 부호화방법.

【청구항 57】

제54항 또는 제55항에 있어서, 상기 제1 내지 제5천공 기저 마스크함수는,

서로 다른 생성다항식에 의해 제1 및 제2 m 시퀀스를 생성하고, 상기 제2 m 시퀀스
를 이용하여 월시부호를 만들기 위한 열처리 함수를 구하며, 상기 열처리 함수의 역함수
에 상기 제1 m 시퀀스를 적용하여 구하여진 마스크함수들 중 기저 마스크함수만을 선택
한 후 첫 번째 심볼과 불특정한 한 심볼을 천공함으로서 생성되는 천공 기저 마스크함수

들입을 특징으로 하는 부호화방법.

【청구항 58】

제54항에 있어서, 상기 입력 데이터가 전송율정보임을 특징으로 하는 부호화방법.

【청구항 59】

제58항에 있어서, 상기 소정 길이가 30임을 특징으로 하는 부호화방법.

【청구항 60】

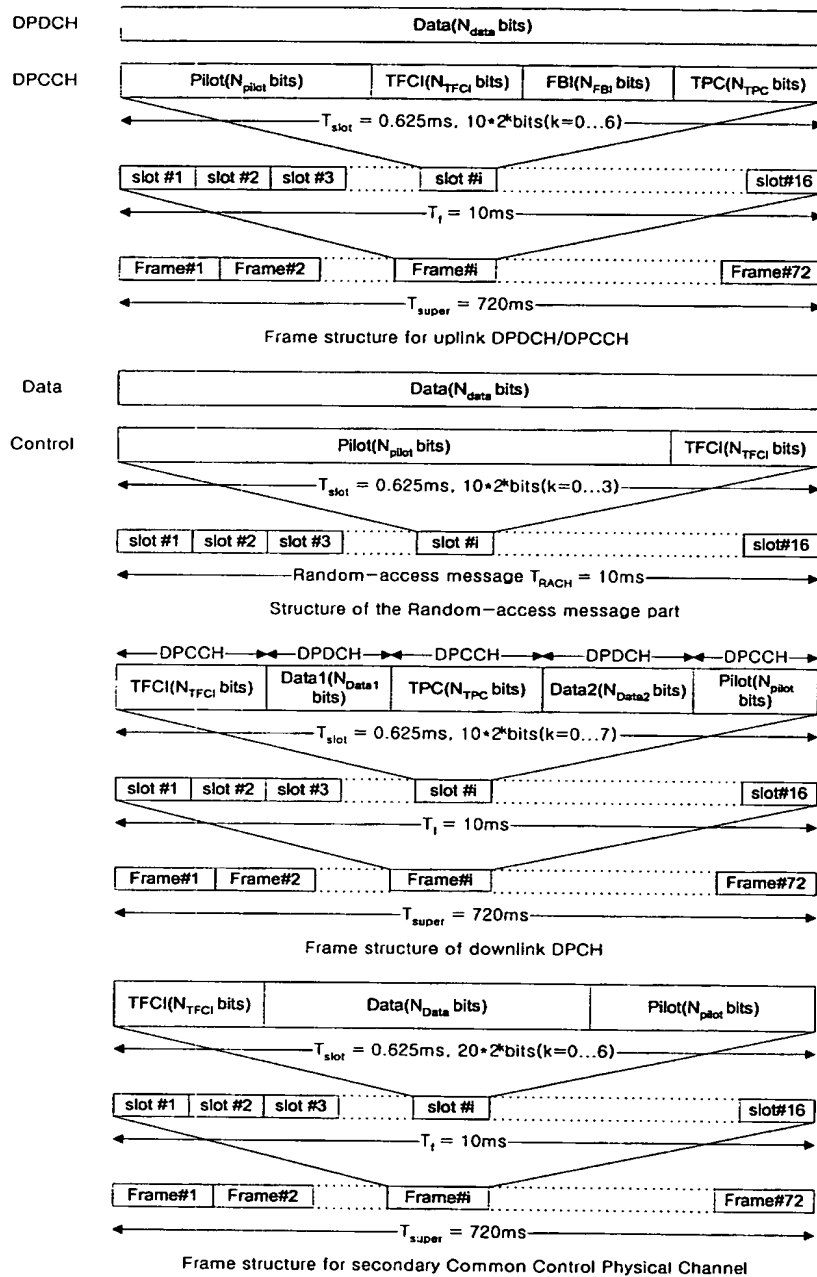
제57항에 있어서, 상기 제1천공 기저 마스크함수는
100100011110011111000101010011임을 특징으로 하는 부호화방법.

【청구항 61】

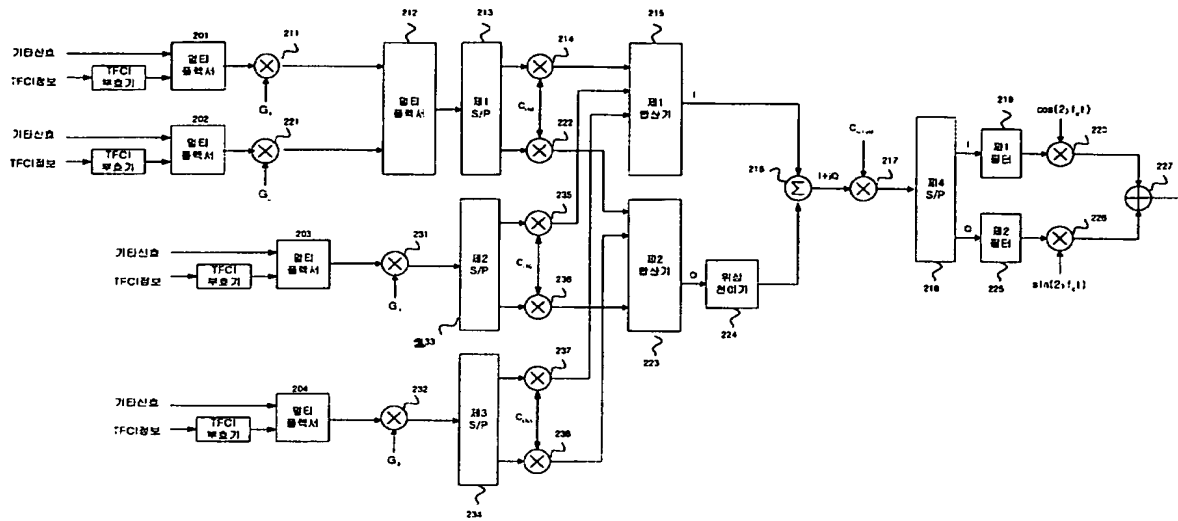
제60항에 있어서, 상기 제2천공 기저 마스크함수는
010100001100011111000001110111이며, 상기 제3천공 기저 마스크함수는
00000111001101110110111000111이며, 상기 제4천공 기저 마스크함수는
000101011111001001101100101011이며, 상기 제5천공 기저 마스크함수는
001110000110111010111101010001임을 특징으로 하는 부호화방법.

【도면】

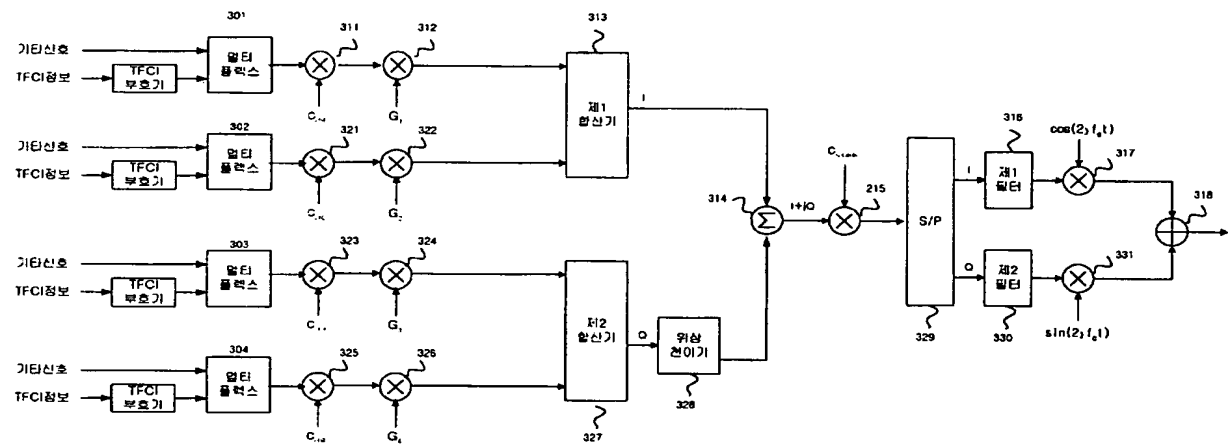
【도 1】



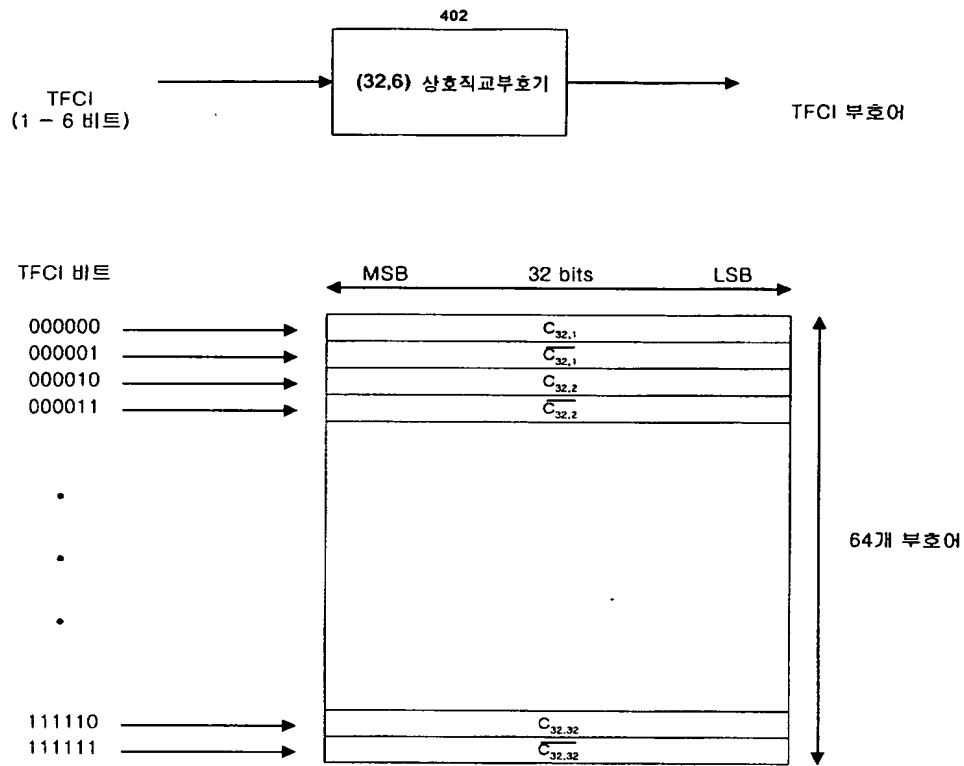
【도 2】



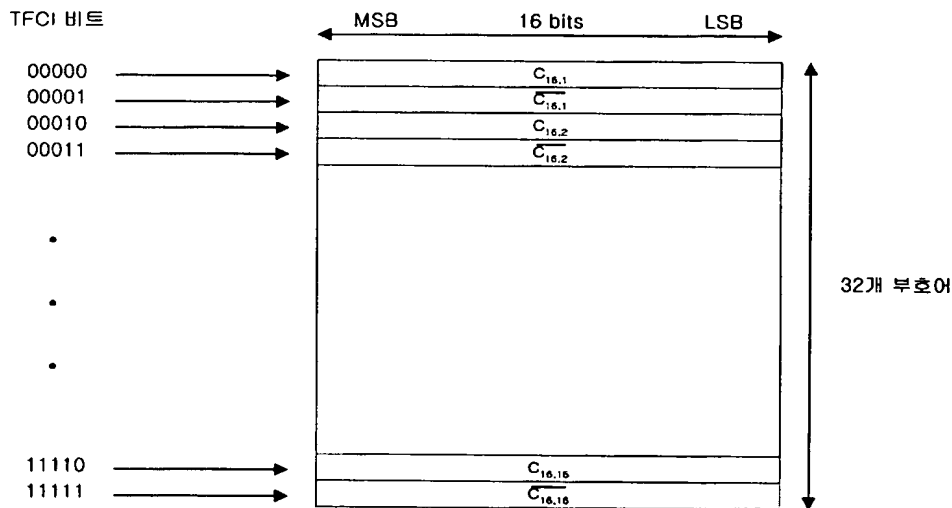
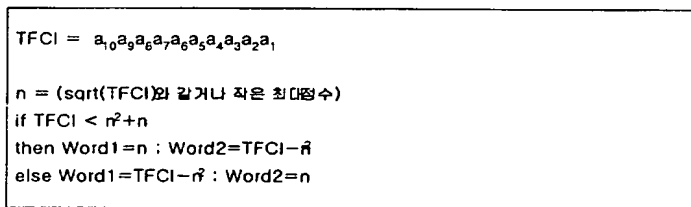
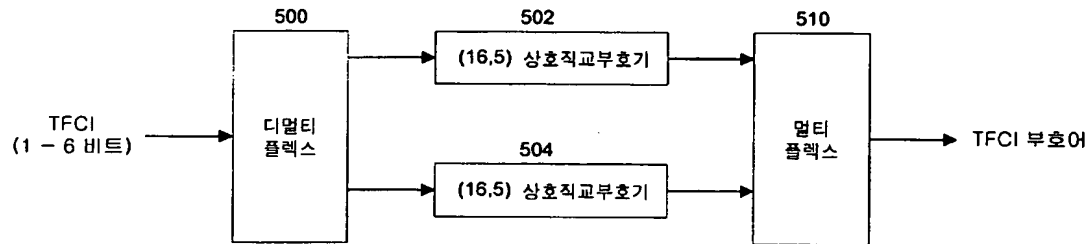
【도 3】



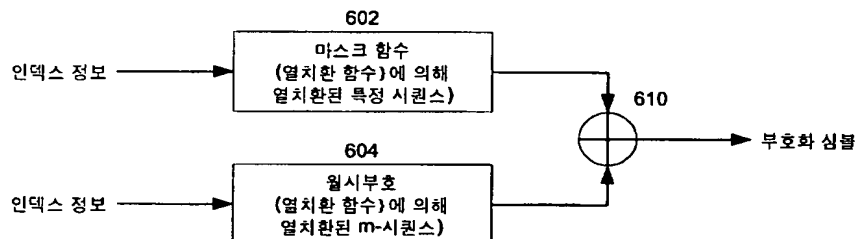
【도 4】



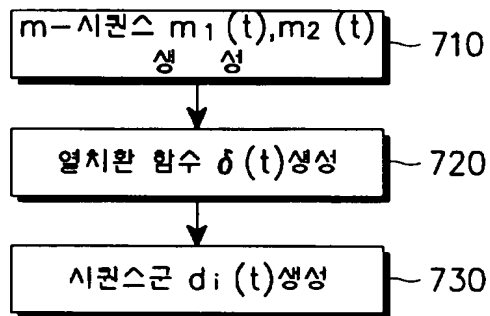
【도 5】



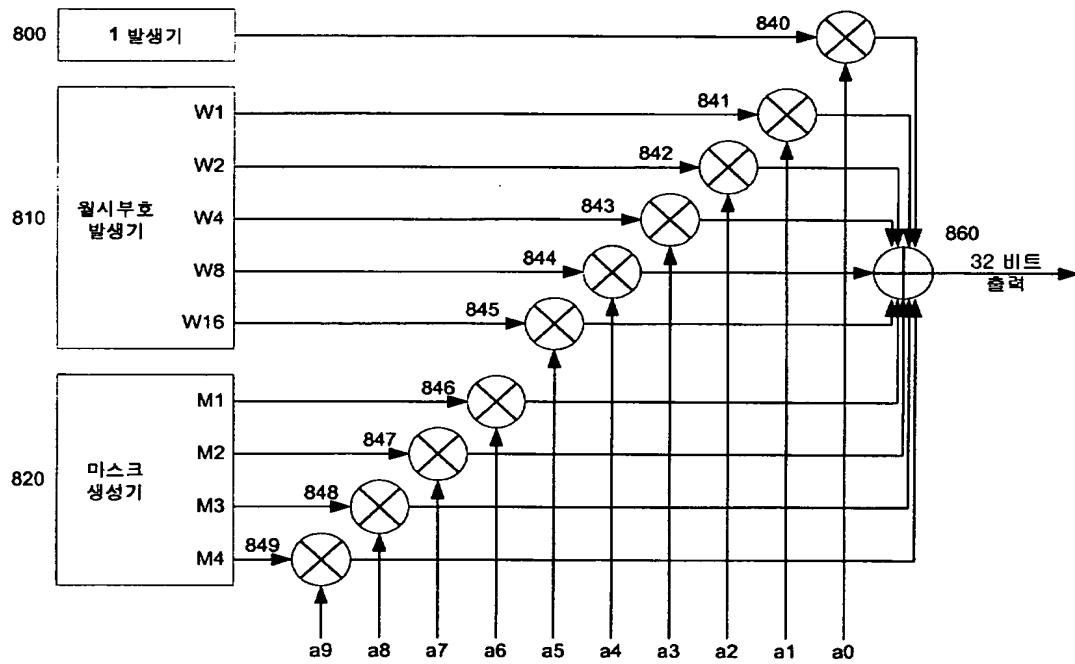
【도 6】



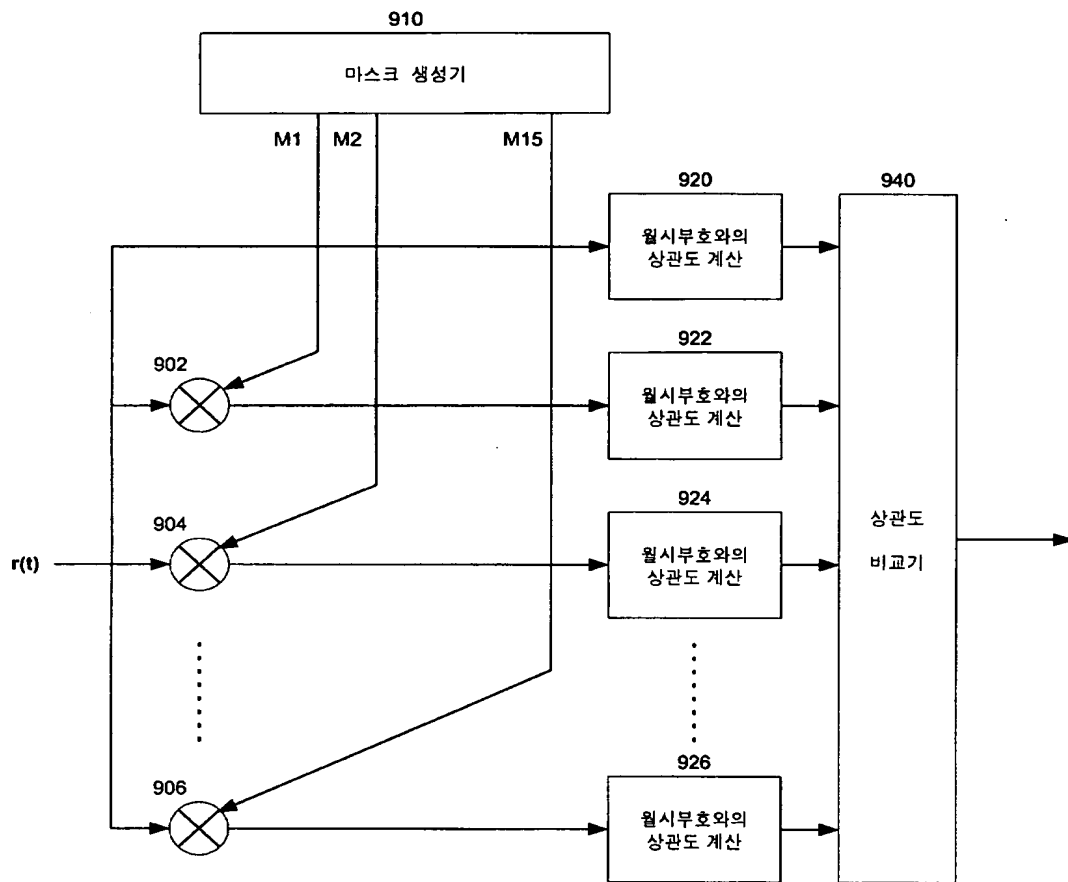
【도 7】



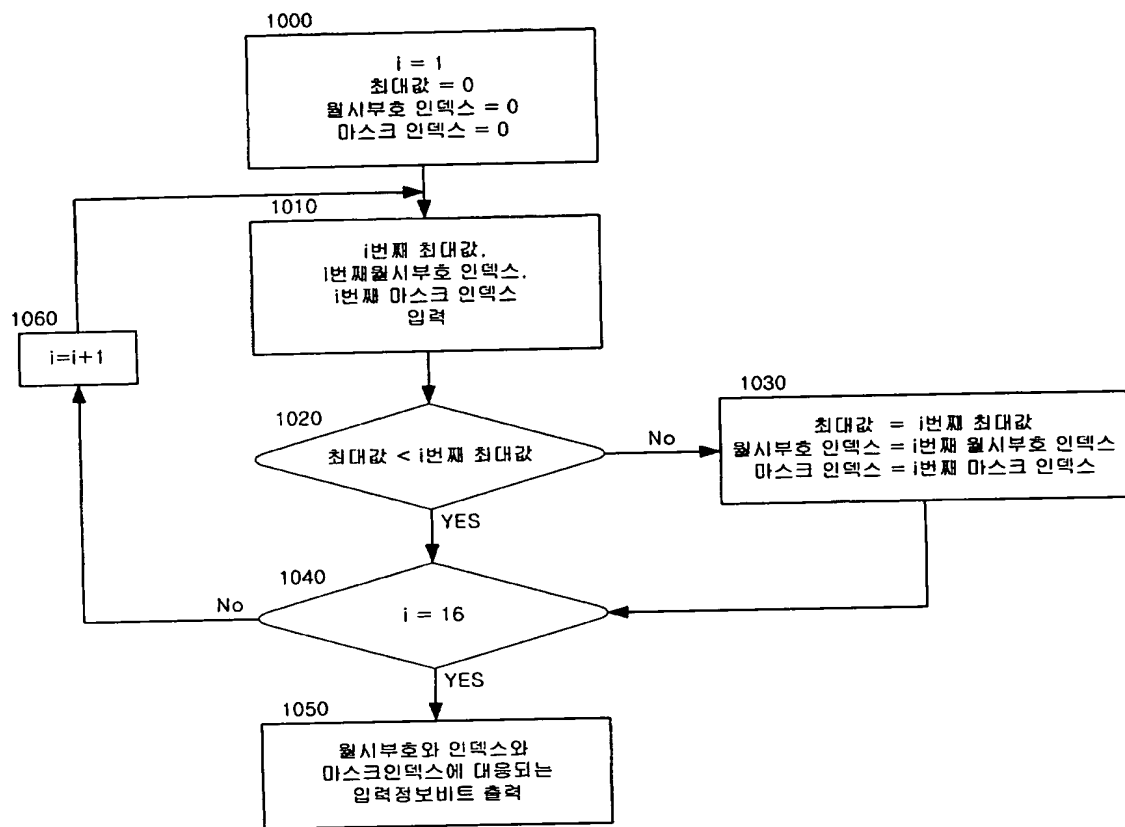
【도 8】



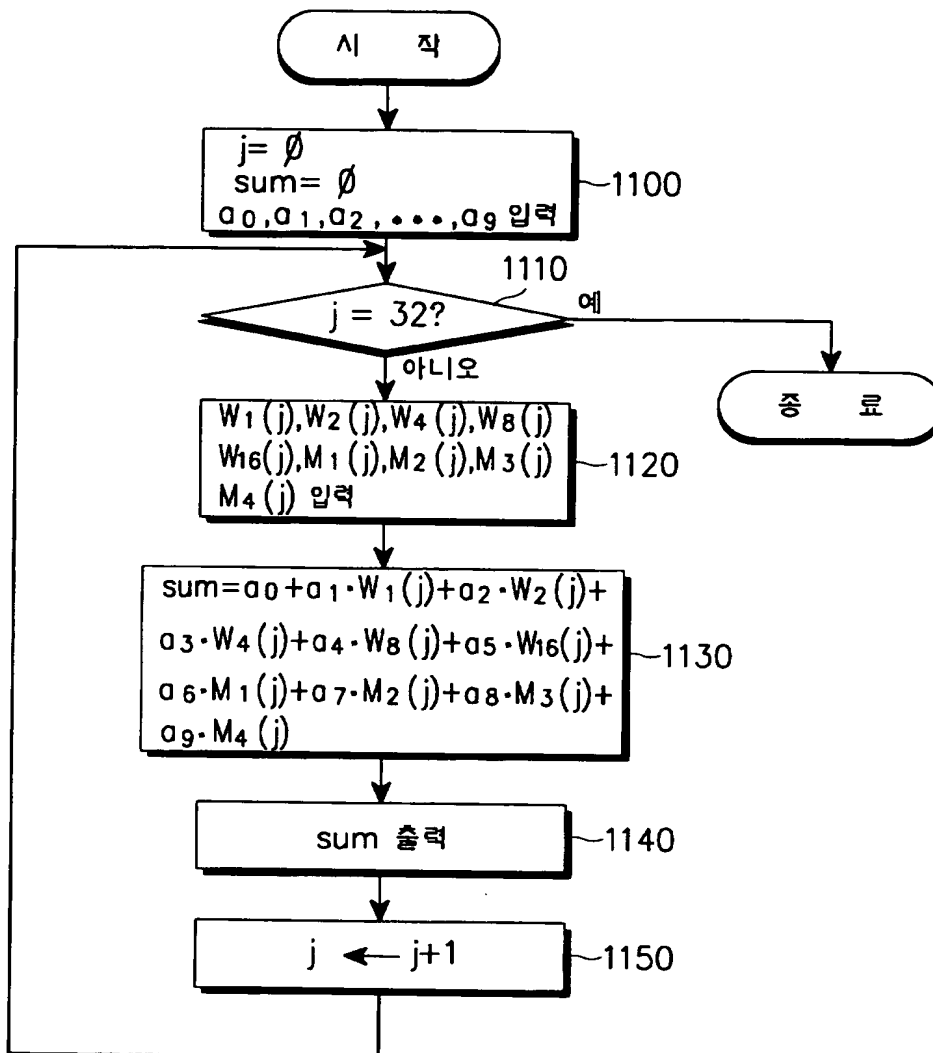
【도 9】



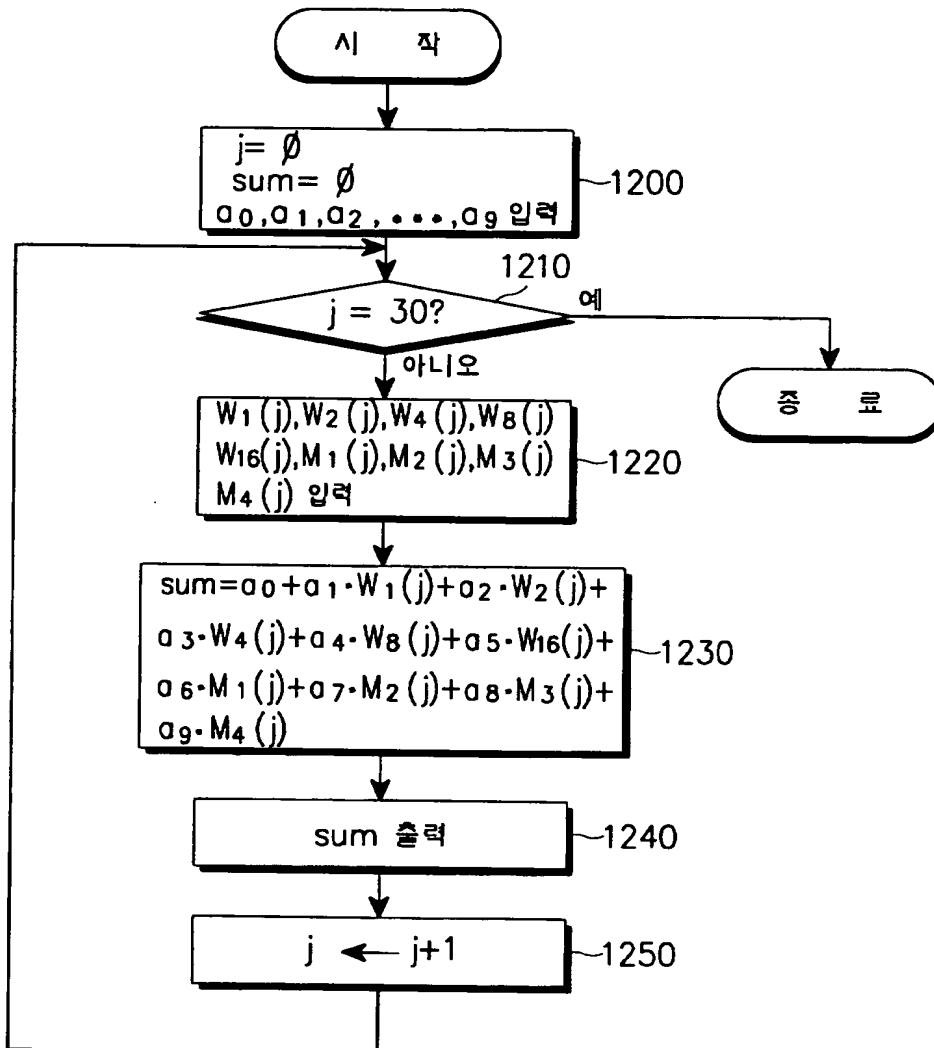
【도 10】



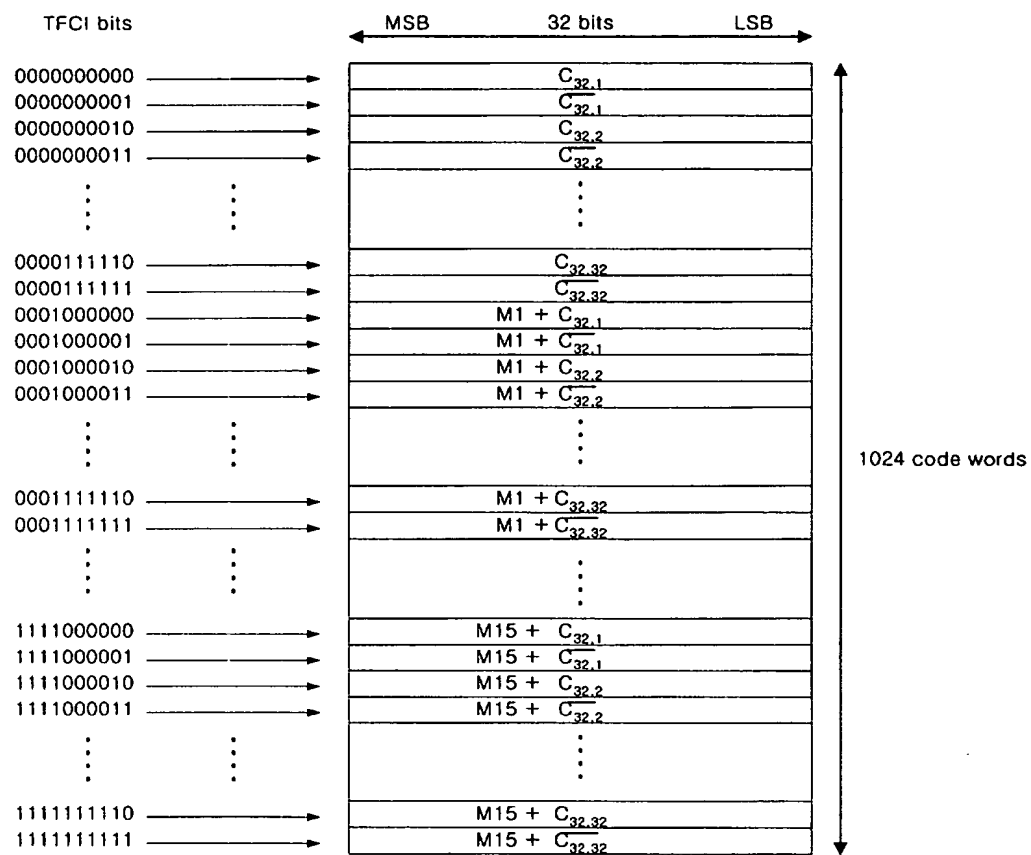
【도 11】



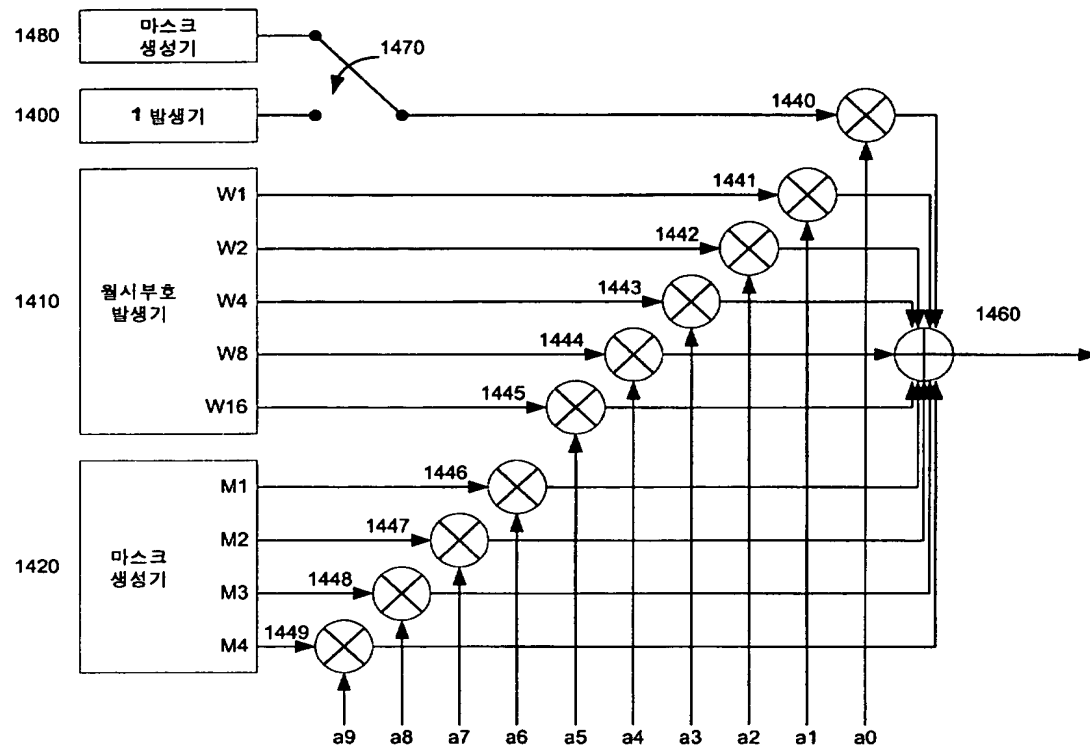
【도 12】



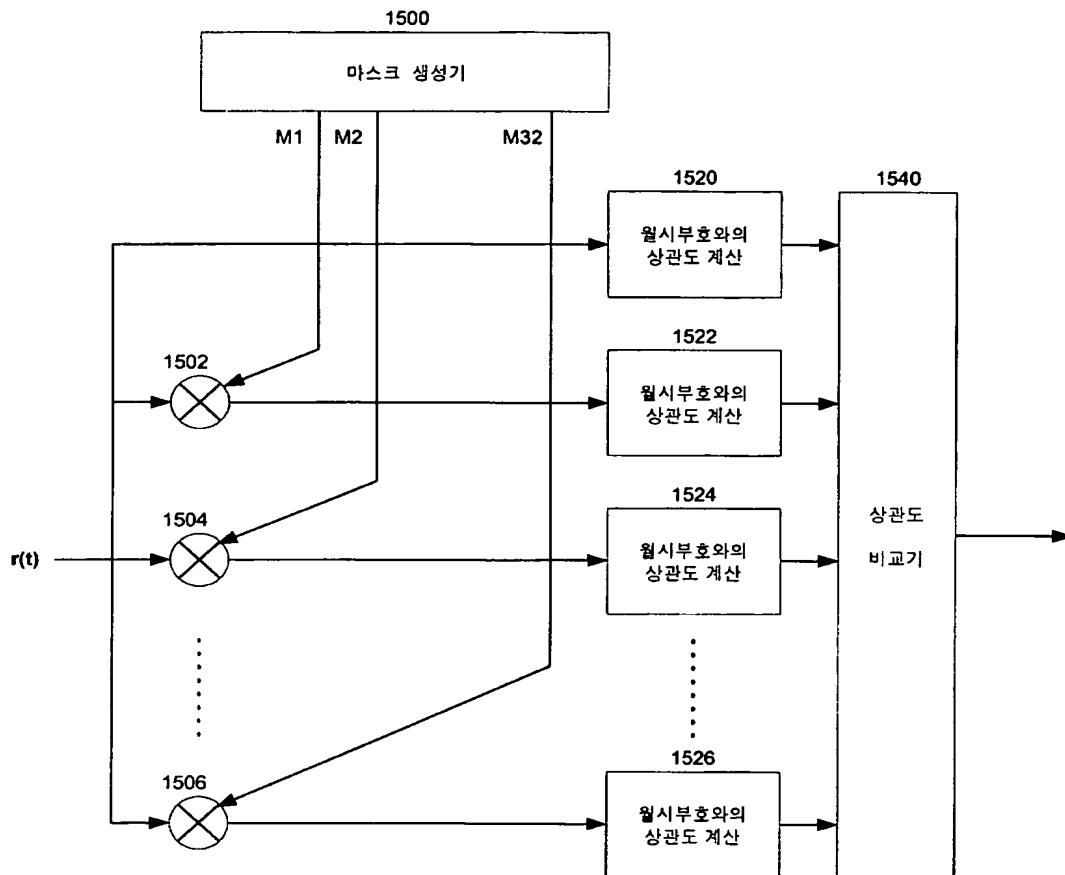
【도 13】



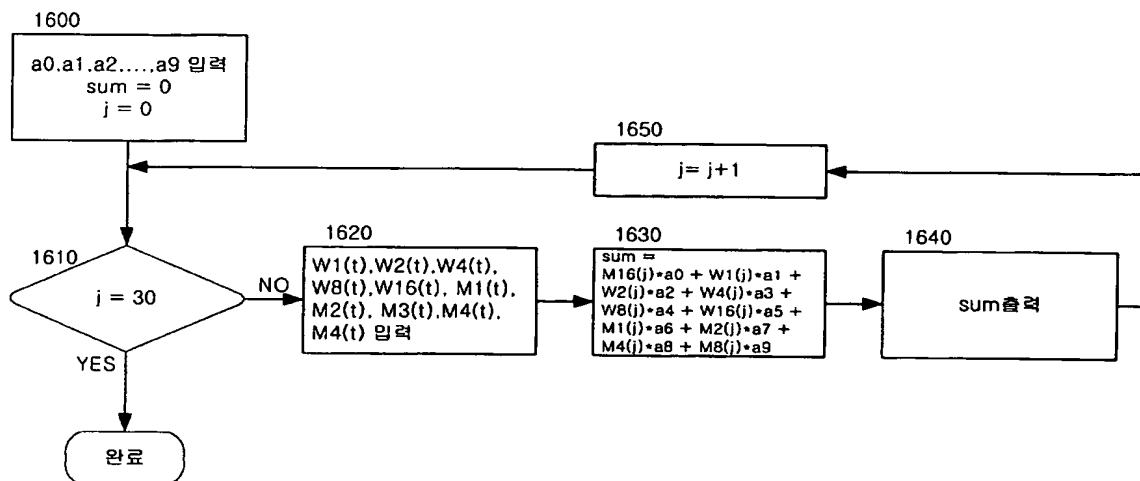
【도 14】



【도 15】



【도 16】



【서류명】	서지사항보정서
【수신처】	특허청장
【제출일자】	1999.08.10
【제출인】	
【명칭】	삼성전자 주식회사
【출원인코드】	119981042713
【사건과의 관계】	출원인
【대리인】	
【성명】	이건주
【대리인코드】	919980003398
【사건의 표시】	
【출원번호】	1019990027932
【출원일자】	1999.07.06
【발명의 명칭】	부호분할다중접속시스템의부호화 /복호화장치및방법
【제출원인】	
【발송번호】	151999001829554
【발송일자】	1999.07.16
【보정할 서류】	특허출원서
【보정할 사항】	
【보정대상 항목】	수수료
【보정방법】	납부
【보정내용】	
【수수료】	미납수수료
【취지】	특허법시행규칙 제13조·실용신안법시행규칙 제12조의 규정에 의하여 위와 같이 제출합니다.
【수수료】	
【보정료】	11000
【기타 수수료】	95000
【합계】	106000